arXiv:2603.01025v1 [cs.LG] 1 Mar 2026

用于推理正确性估计的单令牌验证

Zhan Z huang [1,2] Xiequn Wang [1] Zebin Chen [1] Feiyang Ye [1] Ying Wei [3] Kede Ma [2] Yu Zhang [1]

摘要

大语言模型（大语言模型）近期取得的突破，使其在数学问题求解等复杂推理任务中获得了显著成功。提升性能的一种常见策略是并行思考，即生成多条推理轨迹，并通过多数投票或最佳-N 解码等聚合方案得出最终预测。然而，仍存在两个关键挑战。首先，多样本解码会带来显著的推理延迟，尤其是在生成长篇输出时。其次，可靠评估单条推理轨迹正确性的有效机制仍然有限。为应对这些挑战，我们提出单令牌验证（OTV），这是一种在生成过程中通过单次前向传播估计推理正确性的计算方法。OTV由一个可学习令牌触发，并通过低秩自适应集成到大语言模型中，以探查键值缓存中的内部推理信号，从而支持在生成的任何阶段进行词元级正确性估计，且不会干扰主要推理。在数学推理基准上的实验表明，OTV始终优于现有验证器。此外，通过正确性引导的提前终止，OTV可减少高达90%的词元用量，优先选择更短且更可靠的解答。

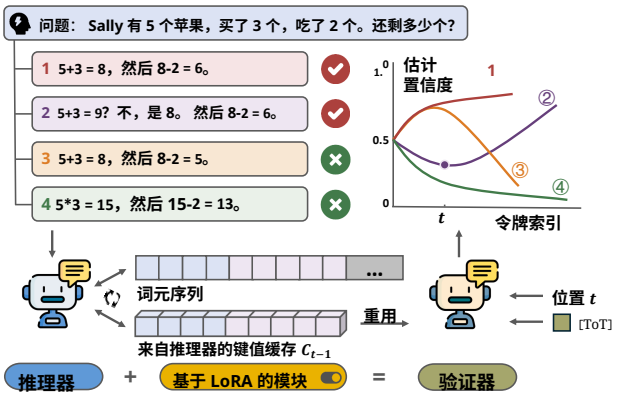


图*1.* 所提出OTV的概念示意图。通过复用键值缓存，并经由特殊词元 ⌈ToT⌋,激活基于LoRA的验证器，OTV能够在单次前向传播中可靠地估计推理轨迹的正确性。

1. 引言

以 OpenAI o1（Jaech 等人，2024）、DeepSeek-R1（Guo 等人，2025年）和 Qwen3 系列（Yang 等人，2025a）为代表的大语言模型，近期展现出强大的多步推理能力，能够应对数学问题求解等具有挑战性的任务。这些进展主要得益于结合了基于人工提供的长思维链进行监督微调（Wei 等人，2022年；Suzgun 等人，2023年）以及利用结果级或过程级反馈进行强化学习（Ouyang 等人，2022年；Shao 等人，2024；Team 等人，2025）的训练流程。与此同时，测试时缩放已成为一种互补范式，可通过在推理期间分配额外计算资源来提高准确率（Brown 等人，2024；Venktesh 等人，2025）。一种简单而有效的方法是并行思考，即模型生成多个候选解答轨迹，并将它们汇总为最终答案。

并行思考面临的一项核心挑战，是如何可靠地评估单条推理轨迹的正确性。现有方法大致可分为内部和外部验证。内部方法依赖模型自身的 token 级不确定性或校准[1]来推导轨迹级置信度[2]分数，进而在多种不同的推理轨迹中进行多数投票（Wang 等人，2023b；Kang 等人，2025；Fu 等人，2025；Zhang 等人，2025a；Huang 等人，2025）。这些技术原生于模型，无需辅助模型，但往往存在校准失准的问题，难以区分正确与错误的推理轨迹，尤其是在长文本解答中（Huang 等人，

[1]南方科技大学，深圳，中国 [2]香港城市大学，香港特别行政区 [3]浙江大学，杭州，中国。通讯作者：Ying Wei< ying.wei@zju.edu.cn>，Kede Ma <kede.ma@cityu.edu.hk>，Yu Zhang <yu.zhang.ust@gmail.com>。

预印本。*2026*年*3*月*3*日。

[1]校准是指预测概率与经验结果频率之间的一致性——例如，在概率为0.7的预测中，大约70%应当是正确的。[2]此处，置信度表示模型对其所生成答案正确的预测概率。

2023；Xiong等人，2024）。外部方法则训练专门的验证器（Cobbe 等人，2021；Hosseini等人，2024；Zhang 等人，2024c；Lightman 等人，2024；Wang 等人，2024；Yang 等人，2024；Zhao 等人，2025），为中间步骤或最终答案打分。尽管这些辅助模型能够提供更丰富的反馈，但它们将基础大语言模型视为黑盒，往往会显著增加推理开销，同时也有领域不匹配的风险。因此，其准确率仍远低于 Pass@k 指标所暗示的上限（Chen 等人，2021）。此外，现有方法通常要等完整的推理轨迹生成后才做出决策，因为其评分规则是针对最终答案或完整轨迹定义的。启用提前终止时，如果中间信号不可靠，性能可能会明显下降。因此，“系统 2”式过度思考（Chen 等人，2024a）会进一步推高解码成本，引发显著的效率问题。

这些局限性要求 一种更深入的内部验证形式，超越表层的、基于logit的启发式方法。这种验证应当：1）明确利用正在进行的推理过程的内部状态（Burns等人，2022年；Azaria与Mitchell，2023年；Zhang 等人，2025a；Li 等人，2025），2）提供细粒度的词元级推理正确性估计，以实现实用的提前终止（Fu 等人，2025；Zhang 等人，2025a；Lee等人，2025），并且3）在生成过程中调用成本低廉。

为此，我们提出单令牌验证（OTV），这是一种通过基于LoRA的验证器增强推理大语言模型的计算方法（Hu 等人，2022年），直接作用于模型的键值（KV）缓存上（见图1）。具体而言，OTV引入了一个特殊的真值令牌（[ToT]）。在推理时插入该令牌后，它利用经过LoRA增强的交叉注意力，访问在持续推理过程中累积的缓存 KV 状态。一个小型回归头将 [ToT] 的最后一层隐藏状态映射为推理正确性的标量估计，从而对部分轨迹提供词元级验证。关键在于，LoRA更新受到门控控制，因此基础大语言模型在默认模式下的行为与原始推理器完全一致；只有在输入中出现 [ToT] 时，模型才会采用验证器角色。因此，OTV复用了大语言模型的完整内部计算，并且每次验证查询仅需一次前向传播。

OTV 的训练完全依赖于为推理轨迹中的每个词元分配的廉价伪分数，这些伪分数源自其最终正确性标签。我们进一步表明，OTV 训练支持并行化：通过在所有候选位置同时插入 [ToT]，并复用缓存前缀，OTV 可在一次前向传播中计算整条轨迹的正确性分数。这保留了标准 Transformer 训练中的令牌级并行性。我们在多个推理大语言模型（即，Qwen3-4B-Instruct、Qwen3-8B，以及 DAPO-Qwen-32B）和一系列数学基准测试上评估了 OTV。在多种并行思考策略及若干高效变体中，OTV 始终优于现有的内部和外部验证器，大幅缩小了与 Pass@k 之间的差距，同时通过置信度引导的提前终止将词元使用量最多减少 90%。除后训练推理大语言模型外，我们还表明，在预训练基础模型上校准 OTV，同样能显著提升原始数学推理准确率和鲁棒性。

2. 相关工作

并行思考 测试时缩放已成为强化大语言模型推理的核心范式，通常沿着三个方向推进：扩大有效输入上下文（例如，检索增强生成（Lewis 等，2020）和工具使用（Schick 等，2023））、增加序贯推理深度（Jaech 等，2024；Muennighoff 等，2025），以及通过并行采样扩大生成规模（Comanici 等，2025；Wen 等，2025；Yang 等，2025b；Hsu 等，2025；Zheng 等，2025b）。其中，并行思考是应用最广泛的方法之一。典型实例包括多数投票（Wang 等，2023b）、Best-of-N 解码（Stiennon 等，2020），以及树结构或图结构搜索（Yao 等，2023；Zhang 等，2024a）。这些方法会探索并评估多个（部分）解答，再确定最终答案；与单轨迹解码相比，它们通常能达到更高的性能上限（Ghosal 等，2025），但代价是产生大量词元开销。为控制这一预算，近期研究探索了轨迹剪枝（Lee 等，2025；Wang 等，2025；Fu 等，2025；Huang 等，2025），以提前终止前景不佳的推理轨迹。

推理正确性估计 评估单条推理轨迹的正确性（Lee& Hockenmaier，2025）对于在并行思考中可靠地提前终止大语言模型至关重要。现有方法可分为内部验证和外部验证。内部方法利用模型自身的表征（Lin 等人，2022年；Fadeeva 等人，2024）。例如，自一致性（Wang 等人，2023b）通过多数投票汇总多样化的推理轨迹，而自确定性（Kang 等人，2025）则从输出分布中推导置信度。DeepConf（Fu 等人，2025）会剪除低置信度的推理轨迹，而自校准（Huang 等人，2025）则将置信度分数提炼为单次推理估计器。基于探测的方法还会在最后一层隐藏状态上训练预测头（Zhang 等人，2025a；Lee等人，2025），这表明模型在内部编码了正确性信号，可用于支持经校准的提前退出。

外部方法则引入辅助验证器，对生成文本进行处理。结果奖励模型仅评估最终答案（Cobbe 等人，2021；Yu 等人，

2023年；Chen 等人，2024b；Liu 等人，2024；Lu 等人，2024；Zhang 等人，2025b），而过程奖励模型则为中间步骤评分，并将其汇总为整体正确性估计（Uesato 等人，2022年；Lightman 等人，2024；Wang 等人，2023a；Zhang 等人，2025c）。后续研究将验证扩展到更广泛的领域（Zeng 等人，2025）和更丰富的标准（Golovneva 等人，2022年；Wang 等人，2024），或将验证重新表述为一种由推理驱动的预测任务（Ankner 等人，2024；Zhang 等人，2024c）。其他研究方向则开发评论器式反馈模型（Zheng 等人，2023年；2025a；Ye 等人，2025），或对多个验证器和解答进行集成（Lifshitz 等人，2025；Zhong 等人，2025；Zhao 等人，2025），以提升可靠性。

尽管外部验证器通常能提供更强的反馈，但它们忽略了基础大语言模型的内部动态，并会引入额外的推理成本，还可能存在领域不匹配。相比之下，内部方法原生于模型且效率较高，但仍容易出现校准失准，尤其是在处理较长解答时。

OTV 处于这两个极端之间。通过附加一个仅在验证模式下激活的基于 LoRA 的模块，OTV 使同一个推理大语言模型也能够充当验证器。它以几乎可以忽略不计的额外计算提供令牌级正确性估计，从而将内部方法的模型特异性和效率，与外部验证器的显式评分能力结合起来。

3. 所提方法：OTV

本节介绍 OTV，这是一种用于推理正确性估计的计算方法。OTV 将推理大语言模型与以下两部分相结合：1）仅在验证期间激活的基于 LoRA 的模块；2）用于探测模型键值缓存的特殊验证令牌。下面介绍其四个组成部分：LoRA 模块设计（第3.1节）、键值缓存探测（第3.2节）、伪标签生成（第 3.3节）以及并行化实现（第3.4节）。

3.1. 基于 LoRA 的验证

LoRA（Hu 等人，2022年）是一种广泛使用的参数高效微调方法，通过低秩更新扩展预训练权重矩阵。给定权重矩阵 $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$，LoRA 引入两个可训练矩阵 $\mathbf{R} \in \mathbb{R}^{d_{\text{out}} \times r}$ 和 $\mathbf{S} \in \mathbb{R}^{r \times d_{\text{in}}}$，其秩为 $r \ll \min\{d_{\text{in}}, d_{\text{out}}\}$，并将更新后的权重定义为 $\mathbf{W}_{\text{LoRA}} = \mathbf{W} + \mathbf{RS}$。LoRA 只训练适配器，从而大幅减少可训练参数数量和优化器状态内存。

在 OTV 中，我们将基于 LoRA 的验证模块连接到基础大语言模型的选定层。为保留原有的推理能力，我们采用了一种门控机制（Samragh 等人，2025），该机制将 LoRA 路径并行添加到每个线性层中，并仅在验证模式下激活。

对于位置 t 的输入 $\boldsymbol{x}_t \in \mathbb{R}^{d_{\text{in}} \times 1}$，相应的输出为

$$\boldsymbol{z}_t = (\mathbf{W} + m_t \, \mathbf{RS}) \, \boldsymbol{x}_t, \tag{1}$$

其中 $m_t \in \{0, 1\}$ 是一个二元门控。当 $m_t = 0$时，大语言模型的行为与原始推理器完全相同；当 $m_t = 1$时，则应用 LoRA 更新，模型承担验证器的角色。门控机制将推理与验证清晰地区分开来，并最大限度地降低微调损害基础模型推理能力的风险。

3.2. 基于 KV 缓存的内部表征探测

在自回归解码过程中，基于 Transformer 的大语言模型会在第 l 层和位置 t 维护一个键值缓存，记作 $\mathcal{C}_t^{(l)} = \{\mathrm{K}_t^{(l)}、\mathrm{V}_t^{(l)}\}$，其中 $\mathrm{K}_t^{(l)} = [\boldsymbol{k}_1^{(l)}, \ldots, \boldsymbol{k}_t^{(l)}]$、$\mathrm{V}_t^{(l)} = [\boldsymbol{v}_1^{(l)}, \ldots, \boldsymbol{v}_t^{(l)}]$和 $\boldsymbol{k}_t^{(l)}、\boldsymbol{v}_t^{(l)} \in \mathbb{R}^{D \times 1}$ 是 D 维的键和值向量，且满足 $\mathcal{C}_0^{(l)} = \emptyset$。对于一个具有 L 层的大语言模型，在位置 $t+1$ 的前向传播由当前输入 $\boldsymbol{x}_{t+1}$ 和累积的键值缓存 $\mathcal{C}_t = \bigcup_{l=1}^{L} \mathcal{C}_t^{(l)}$ 函数性地决定。与最后一层隐藏状态（Zhang 等，2025a；Lee 等，2025）相比，后者只是对前序上下文的有损概括，并丢弃了其逐词元和逐层结构，而键值缓存则是底层大语言模型前缀的充分统计量，可对完整推理轨迹进行更丰富、任务特定的池化。

OTV 旨在显式利用这一结构。我们引入一个特殊的真理词元([TOT])，它仅在验证模式下插入（即，当 $m_{t+1} = 1$ 在方程（1）中），位置任意 $t+1$，用于估计截至词元 t 的部分轨迹的正确性。模型无需重新计算前缀，而是复用每一层的键值缓存 $\mathcal{C}_t$，并执行一次前向传播，同时加入基于 LoRA 的验证模块，详见第3.1节。

令 $\boldsymbol{x}_{t+1}^{(l)}$ 为第 l层的输入，其中 $\boldsymbol{x}_{t+1}^{(0)}$ 表示 [ToT]的初始嵌入。经 LoRA 增强的查询、键和值向量定义为

$$\begin{cases} \tilde{\boldsymbol{q}}_{t+1}^{(l)} = \left(\mathbf{W}_q^{(l)} + \mathbf{R}_q^{(l)} \mathbf{S}_q^{(l)}\right) \boldsymbol{x}_{t+1}^{(l)}, \\ \tilde{\boldsymbol{k}}_{t+1}^{(l)} = \left(\mathbf{W}_k^{(l)} + \mathbf{R}_k^{(l)} \mathbf{S}_k^{(l)}\right) \boldsymbol{x}_{t+1}^{(l)}, \\ \tilde{\boldsymbol{v}}_{t+1}^{(l)} = \left(\mathbf{W}_v^{(l)} + \mathbf{R}_v^{(l)} \mathbf{S}_v^{(l)}\right) \boldsymbol{x}_{t+1}^{(l)}. \end{cases} \tag{2}$$

其中 $\mathbf{W}_*^{(l)}, * \in \{q, k, v\}$表示第 l 层冻结的预训练权重，而 $\mathbf{R}_*^{(l)}, \mathbf{S}_*^{(l)}$是对应的可训练 LoRA 矩阵。利用缓存前缀 $\mathcal{C}_t$ 和更新后的向量 $\tilde{\boldsymbol{q}}_{t+1}^{(l)}, \tilde{\boldsymbol{k}}_{t+1}^{(l)}, \tilde{\boldsymbol{v}}_{t+1}^{(l)}$，第 l 层中[ToT] 的注意力输出为

$$\tilde{\boldsymbol{h}}_{t+1}^{(l)} = \left[\mathbf{V}_t^{(l)}, \, \tilde{\boldsymbol{v}}_{t+1}^{(l)}\right] \operatorname{softmax}\left(\frac{\left[\mathbf{K}_t^{(l)}, \tilde{\boldsymbol{k}}_{t+1}^{(l)}\right]^{\intercal} \tilde{\boldsymbol{q}}_{t+1}^{(l)}}{\sqrt{D}}\right), \tag{3}$$

其中 softmax(·) 按列应用。[ToT] 经过所有 L 层传播后，我们得到其最终隐藏状态

$\tilde{\boldsymbol{h}}_{t+1}^{(L)} \in \mathbb{R}^{D\times 1}$。OTV 应用一个三层感知机$g(\cdot)$，将 $\tilde{\boldsymbol{h}}_{t+1}^{(L)}$ 映射为标量预测：

$$\hat{c}_t = g\left(\tilde{\boldsymbol{h}}_{t+1}^{(L)}\right) \in [0,1], \tag{4}$$

用于估计推理轨迹截至位置 t正确的可能性（即，置信度）。预测连续分数，而不是从词汇表中解码词元，可以避免验证与模型对特定词元（例如，“正确”和“错误”）的语言先验相互纠缠，并产生更易于校准和设置阈值以进行路由的信号。

3.3. 令牌级伪置信度标注

OTV 需要令牌级监督来训练验证器，但收集此类过程监督信号（Lightman 等人，2024）或生成基于搜索的展开代价高昂（Wang 等人，2023a；Luo 等人，2024；Zhang 等人，2024b；Feng 等人，2024；Setlur 等人，2024；Guan 等人，2025）。因此，我们仅根据结果级监督推导伪置信度目标。具体而言，给定一个训练数据集，我们采样一条推理轨迹 $\boldsymbol{x}_{1:T} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$，并在每个词元位置 t指定一个标量目标 $c_t \in [0,1]$。我们将 $c_t = 0$ 解释为确信错误，将$c_t = 1$ 解释为确信正确，将 $c_t = 0.5$ 解释为最大程度的不确定。令 $y \in \{0,1\}$ 表示整条轨迹的最终正确性。在默认设置下，我们将 $c_0 = 0.5$ 初始化为中性先验，并随着证据累积，逐步将置信度偏向y。我们通过线性斜坡实现这一点，使不确定性到最终结果之间进行单调线性插值：

$$c_t = \mathrm{linear}(t) = 0.5 + (y - 0.5)\frac{t}{T}. \tag{5}$$

分配令牌级伪置信度标签后，我们通过最小化响应令牌上的均方误差（MSE），来训练基于 LoRA 的验证模块和回归头：

$$\ell = \frac{1}{T}\sum_{t=1}^{T}(c_t - \hat{c}_t)^2. \tag{6}$$

虽然我们采用线性斜坡作为默认的伪置信度先验，但我们也探索了几种替代标注规则（例如：常数、Sigmoid、噪声扰动和阶梯式斜坡）。此外，我们在第4.4 节和附录C中分别详细介绍了消融研究，以评估性能，并对伪置信度标注规则进行理论分析。

3.4. 并行化

Transformer 的一项关键优势（Vaswani 等人，2017）是能够通过令牌级并行进行训练。OTV 保留了这一特性：尽管位置 t 的置信度分数是通过单令牌验证获得的（见公式（3）），但我们可以通过并行探测所有前缀，仅用一次前向传播就计算出轨迹中所有位置的分数。

考虑一条长度为 T 的推理轨迹。对于每个以 $t \in \{1,\ldots,T\}$ 结尾的前缀，我们希望得到一个验证器查询，该查询仅关注与该前缀相关联的缓存的键值 $\mathcal{C}_t$（即，前 t 个词元）。为此，我们构造一个由 $T+1$ 个真值令牌组成的探针序列，$[\mathrm{ToT}]_{1:T+1}$。在验证模式下，我们收集其在第l 层的 LoRA 增强投影，分别记为 $\widetilde{\mathbf{Q}}_{T+1}^{(l)}$、$\widetilde{\mathbf{K}}_{T+1}^{(l)}$、$\widetilde{\mathbf{V}}_{T+1}^{(l)} \in \mathbb{R}^{D\times(T+1)}$。为确保探针令牌 $t+1$ 只能“看到”截至 t 的前缀，我们使用三角掩码 $\mathbf{M} \in \mathbb{R}^{T\times(T+1)}$，其定义如下：

$$M_{i,j} = \begin{cases} 0, & i < j, \\ -\infty, & i \geq j, \end{cases} \tag{7}$$

其中 $i \in \{1,\ldots,T\}$ 和 $j \in \{1,\ldots,T+1\}$。在第 l 层，缓存到探针注意力对数值为

$$\mathbf{A}_{\mathrm{ctp}}^{(l)} = \left(\mathbf{K}_T^{(l)}\right)^{\intercal} \widetilde{\mathbf{Q}}_{T+1}^{(l)} + \mathbf{M}. \tag{8}$$

为匹配单令牌计算（见公式 (3)），每个探针令牌在探针令牌集合内只能关注自身。因此，我们将每个探针的自对数值计算为探针间点积矩阵的对角线元素：

$$\boldsymbol{a}_{\mathrm{self}}^{(l)} = \mathrm{diag}\left(\left(\widetilde{\mathbf{K}}_{T+1}^{(l)}\right)^{\intercal} \widetilde{\mathbf{Q}}_{T+1}^{(l)}\right) \in \mathbb{R}^{(T+1)\times 1}, \tag{9}$$

其中，diag(·) 提取方阵的对角线元素，并将其作为列向量返回。随后，我们将这些自对数值追加为一行，并沿列应用 Softmax，以获得注意力概率：

$$\begin{aligned} \mathbf{A}_{T+1}^{(l)} &= \begin{bmatrix} \mathbf{A}_{\mathrm{ctp}}^{(l)} \\ \left(\boldsymbol{a}_{\mathrm{self}}^{(l)}\right)^{\intercal} \end{bmatrix} \in \mathbb{R}^{(T+1)\times(T+1)}, \\ \mathbf{P}_{T+1}^{(l)} &= \begin{bmatrix} \mathbf{P}_{\mathrm{ctp}}^{(l)} \\ \left(\boldsymbol{p}_{\mathrm{self}}^{(l)}\right)^{\intercal} \end{bmatrix} = \mathrm{softmax}\left(\frac{\mathbf{A}_{T+1}^{(l)}}{\sqrt{D}}\right). \end{aligned} \tag{10}$$

第 l 层所有探针对应的注意力输出为

$$\widetilde{\mathbf{H}}_{T+1}^{(l)} = \mathbf{V}_T^{(l)}\,\mathbf{P}_{\mathrm{ctp}}^{(l)} + \widetilde{\mathbf{V}}_{T+1}^{(l)}\,\mathrm{diag}\left(\boldsymbol{p}_{\mathrm{self}}^{(l)}\right) \in \mathbb{R}^{D\times(T+1)}. \tag{11}$$

这里，diag(·) 将列向量转换为方形对角矩阵。最后，将这些计算沿所有L 层堆叠起来，得到 $T+1$ 最终的 [ToT] 隐藏状态，每个状态都映射为一个标量置信度（见公式（4））。从数学上看，这一构造等同于运行 $T+1$个独立的单词元验证查询。不过，它通过复用同一缓存前缀，将这些查询整合到一次前向传播中。我们仅在对应于响应词元的探测位置计算均方误差损失，不包括提示和查询片段中的位置。训练和推理流程的算法描述见附录A。

4. 实验

本节中，我们在标准和高级数学推理场景下，测试 OTV 在多个开源推理大语言模型上的并行思考能力。我们考察常见的聚合方案，包括自一致性、加权自一致性和 best-of-N，以及提前终止的变体。我们报告准确率和效率指标（即，令牌成本和验证开销），并定性分析 OTV 的置信度轨迹随生成过程的变化。

4.1. 实验设置

模型与数据集 我们在三种具有代表性的开源推理大语言模型上评估 OTV：*Qwen3-4B-Instruct*、*Qwen3-8B* (Yang 等人，2025a) 以及 *DAPO-Qwen-32B* (Yu 等人，2025)。在所有实验中，基础大语言模型均保持冻结，并用作推理器。为涵盖不同难度的推理任务，我们设置了两种实验方案。对于标准数学推理，我们在 MetaMathQA 上训练 OTV（Yu 等人，2024），并在 GSM8K 上测试（Cobbe 等人，2021），后者是广泛采用的小学数学题基准测试。对于高级数学推理，我们在 DAPO17K 上训练（Yu 等人，2025），这是一个难度更高的数据集，仅提供最终答案（不含人工编写的推理轨迹）；随后在 AIME24（MAA，2024）和 AIME25（MAA，2025）基准测试上进行评估。这两个基准测试均由美国数学邀请赛的竞赛风格问题组成。我们的主要实验聚焦于这一高级方案，因为它能严格检验长程、多步逻辑推理能力。

训练流程 对于每个训练实例，我们对基础大语言模型进行 8 次采样，以获得多样化的推理轨迹，并根据最终正确性标签推导出令牌级伪置信度目标，用于监督 OTV。在推理时，对于每个测试问题，我们首先生成一个包含 256 条完整推理轨迹的集合，并将温度参数设为 1。0，然后反复从该集合中采样子集，以进行后续聚合。除非另有说明，我们使用 `LlamaFactory` (Zheng 等人，2024) 微调 LoRA 模块（应用于查询、键和值投影）以及回归头，训练 3 轮，学习率为 10^{-4}，批次大小为 128。

竞争方法 我们将竞争方法分为内部验证器和外部验证器。内部方法是模型原生的方法，包括 DeepConf（Fu 等人，2025），一种基于词元级对数似然、无需训练的置信度估计器；以及 GenRM（Zhang 等人，2024c），一种推理诱导验证器，它向模型本身提示“答案正确吗？”，并将模型预测“Yes/No”的概率作为置信度分数。外部方法采用现成的奖励模型，包括 AceMath-RM-7B（Liu 等人，2024），一种结果级奖励模型，用于对完整解答进行评分；以及若干过程奖励模型——VersaPRM（Zeng 等人，2025）、Math-Shepherd-Mistral-7B（Wang 等人，2023a）和 Qwen2.5-MathPRM 变体（Zhang 等人，2025c）——沿推理轨迹为各个步骤分配奖励。遵循以往做法，我们将最后一个词元处预测的奖励作为验证器的总体评分。

推理时聚合策略 我们评估了三类并行思考解码器。

- **自一致性** (即，多数投票）及其加权变体（Wang 等人，2023b），其中每条推理轨迹的最终答案均根据其估计置信度分数加权。
- **最佳-N**，从 N 个候选中选取得分最高的轨迹。
- **高效最佳-N 变体**，通过剪枝或提前终止来减少计算：
  - *Drop*@10：周期性基于分数的剪枝。每生成 10 个词元，就丢弃当前得分最低的轨迹，并继续解码，直到只剩下一条轨迹。– *Stop*@600：固定长度的提前确定。当轨迹达到 600 个词元后，终止除得分最高轨迹以外的所有轨迹，并仅继续生成幸存轨迹，直至完成。– *Halve*@300：分阶段减半。每生成 300 个词元，就根据当前分数移除得分处于后半部分的轨迹，并重复这一“减半并继续”的过程，直到只剩下一条轨迹。

我们还报告 Pass@1 （单轨迹准确率）、Pass@k（在 k 条轨迹中的预言机成功率）以及 Maj@k （对 k 条轨迹进行无权重多数投票所得的准确率）。值得注意的是，Pass@k是任何在推理时无法使用真实标注的聚合方法的上限。自一致性和标准最佳-N 都需要以相近的词元成本完成所有轨迹，而高效最佳-N变体则能大幅减少解码量。例如，当 $N = 128$ 且轨迹平均长度为 6,000 个词元时，基于剪枝的变体可将总生成量减少近90%。另一方面，OTV 带来的计算和词元开销可以忽略不计，因为每个验证查询都通过一次前向传播实现（见附录B.4）。

4.2. 主要结果

我们首先在离线设置下评估 OTV 在高级数学推理任务中的表现，此时可以获取完整的采样推理轨迹集合，并在事后进行聚合。表1 报告了使用128 条轨迹进行每次运行时的加权多数投票结果。我们根据轨迹级别的

表 *1.* AIME 上的加权多数投票准确率（%）。对于过程奖励模型，我们使用最终令牌分数作为每条推理轨迹的置信度。遵循 Fu 等人 (2025)，我们采用针对不同方法的聚合窗口：置信度按每条轨迹最后若干个词元的平均分计算（OTV 使用 100 个词元；DeepConf 使用 2,048 个词元）。随后，在进行加权多数投票前，按置信度剔除排名靠后的 50% 推理轨迹。每次运行采样 128 条推理轨迹，所有结果均为 64 次运行的平均值。我们还报告 Pass@128，作为并行思考的预言机上界。每列中，最佳结果以粗体标出，次佳结果加下划线。

		QWEN3-4B		QWEN3-8B		DAPO-QWEN-32B	
		AIME24	AIME25	AIME24	AIME25	AIME24	AIME25
	PASS@128	91.46	83.13	69.58	56.98	83.75	68.75
Internal	PASS@1	60.29	46.67	26.29	19.32	51.77	36.42
	MAJ@128	75.42±1.61	66.46±1.16	44.22±2.11	28.44±2.06	66.72±0.93	41.77±1.76
	DEEPCONF	77.76±2.43	66.77±1.18	45.73±1.50	31.87±2.35	66.77±1.17	44.79±1.65
	GENRM	79.11±2.06	66.72±1.82	44.38±2.27	32.66±2.30	66.72±0.59	42.55±1.66
External	ACEMATH-RM-7B	67.66±1.74	60.00±2.36	44.90±2.70	26.72±2.18	61.15±2.22	42.76±2.00
	VERSAPRM-8B	75.52±1.69	66.72±1.10	44.48±2.30	29.11±2.22	66.98±0.97	43.18±1.81
	MATH-SHEPHERD-7B	74.27±1.50	66.56±1.02	44.22±1.79	26.04±1.94	65.73±2.24	43.65±1.93
	QWEN2.5-PRM800K-7B	75.16±1.66	66.35±1.14	44.22±2.30	29.32±2.57	67.03±1.46	48.18±3.48
	QWEN2.5-PRM-7B	78.18±1.85	63.49±2.39	45.10±2.35	29.48±2.63	67.45±2.26	46.77±3.06
	OTV（我们的方法）	**83.33**±1.57	**69.32**±1.46	**46.56**±1.25	**33.85**±1.69	**70.83**±1.86	**49.58**±1.10

表 *2.* 在 AIME 上、$N = 128$ 条件下 Best-of-N 及三种计算高效变体（即，Drop@10、Stop@600 和 Halve@300）的准确率（%）和平均输出长度（括号内）。所有结果均为 64 次运行的平均值。

	BEST-OF-N		DROP@10		S顶部@600		HALVE@300	
	AIME24	AIME25	AIME24	AIME25	AIME24	AIME25	AIME24	AIME25
				QWEN3-4B				
DEEPCONF	64.95（9664）	43.07（9322）	62.86（7044）	40.78（7260）	59.90（7150）	40.89 (7084)	61.98 (6742)	43.54 (6318)
VERSAPRM-8B	54.48 (6560)	43.28 (6132)	60.52 (2589)	43.44 (6334)	63.75 (3270)	46.77 (6113)	59.06 (2981)	37.24 (6438)
MATH-SHEPHERD-7B	**73.59** (5820)	**66.51** (5824)	61.61 (5679)	46.25 (6116)	57.45 (5989)	47.45 (6218)	54.64 (5722)	45.36 (6118)
QWEN2.5-PRM800K-7B	69.90 (4891)	45.10 (6202)	**66.77** (2196)	42.86 (6445)	60.31 (3434)	**49.53** (6409)	62.24 (2919)	45.21 (6619)
QWEN2.5-PRM-7B	71.77 (3720)	53.33 (3948)	63.80 (3040)	45.83 (6588)	**65.73** (4211)	44.95 (6304)	66.46 (3173)	45.73 (6416)
OTV（我们的方法）	73.44 (5447)	53.91 (5416)	63.39 (4427)	**46.46** (3225)	63.75 (4431)	49.11 (6542)	**67.03** (4132)	**49.02** (3170)
				DAPO-QWEN-32B				
DEEPCONF	53.92 (5101)	38.91 (3957)	50.52 (6382)	37.08 (4398)	51.82 (7176)	37.76 (4353)	50.94 (3772)	36.77 (4449)
VERSAPRM-8B	48.80 (5061)	31.04 (4447)	**59.79** (5263)	39.48 (4796)	53.12 (5432)	37.66 (5005)	49.32 (5744)	36.61 (5046)
MATH-SHEPHERD-7B	62.34 (5051)	42.40 (4570)	55.52 (4819)	39.22 (4475)	55.21 (5555)	36.46 (5063)	**58.80** (4983)	42.76 (4919)
QWEN2.5-PRM800K-7B	54.17 (4722)	**47.81** (4426)	53.28 (5585)	41.20 (4689)	49.48 (5260)	40.89 (4523)	47.40 (5732)	40.00 (4528)
QWEN2.5-PRM-7B	57.03 (4888)	47.24 (4481)	55.52 (5525)	33.65 (5351)	**55.57** (4939)	35.31 (5068)	51.98 (5660)	36.09 (4967)
OTV（我们的方法）	**63.18** (4623)	47.08 (4079)	55.95 (3397)	**50.68** (2926)	53.54 (3211)	**48.23** (2577)	55.05 (3436)	**46.98** (2991)

置信度分数为投票加权，并在聚合前丢弃置信度最低的 50% 轨迹。

在所有三种骨干模型规模下，OTV 在 AIME24 和 AIME25 上始终取得最高准确率。与无权多数投票相比，提升幅度显著（例如，Qwen3-4B 在 AIME24 上提升 +7.9 分，DAPO-Qwen-32B 在 AIME25 上提升 +7.8 分），这表明，与启发式置信度基线相比，OTV 的置信度估计与推理轨迹正确性的相关性更强。与其他内部方法相比，OTV 也始终表现更优，这说明，通过探测模型内部状态的学习型验证器，相较于基于 logits 的置信度或自我询问式“是/否”验证，能够提供更可靠的正确性信号。此外，尽管外部验证器经过了大量训练且应用广泛，但在所有设置下仍不及 OTV，这凸显了将通用奖励模型迁移到不同骨干模型和不断变化的推理轨迹分布上的难度。

表2 比较了 OTV 与标准 Best-of-N 及三种高效变体（即，Drop@10、Stop@600和 Halve@300）在 $N = 128$ 的在线设置下的表现。一项关键发现是，原版 Best-of-N 在准确率方面可能并非最优：在各任务上的平均表现中，其准确率比加权多数投票（解码成本相同）低超过 10 个百分点，这反映出仅凭最高预测置信度分数来选择单条推理轨迹的做法十分脆弱。

在各种高效变体中，OTV 在大多数配置下都能达到最佳或接近最佳的准确率。此外，与竞争方法不同，它通常比标准 Best-of-N 产生更短的最终推理轨迹（平均少约 20% 的词元）。这种行为与 OTV 线性递增的伪置信度目标一致：给定两条正确轨迹，较短的轨迹会更快积累置信度，更早达到更高的分数，因此更有可能在剪枝或提前停止标准下保留下来。在高效

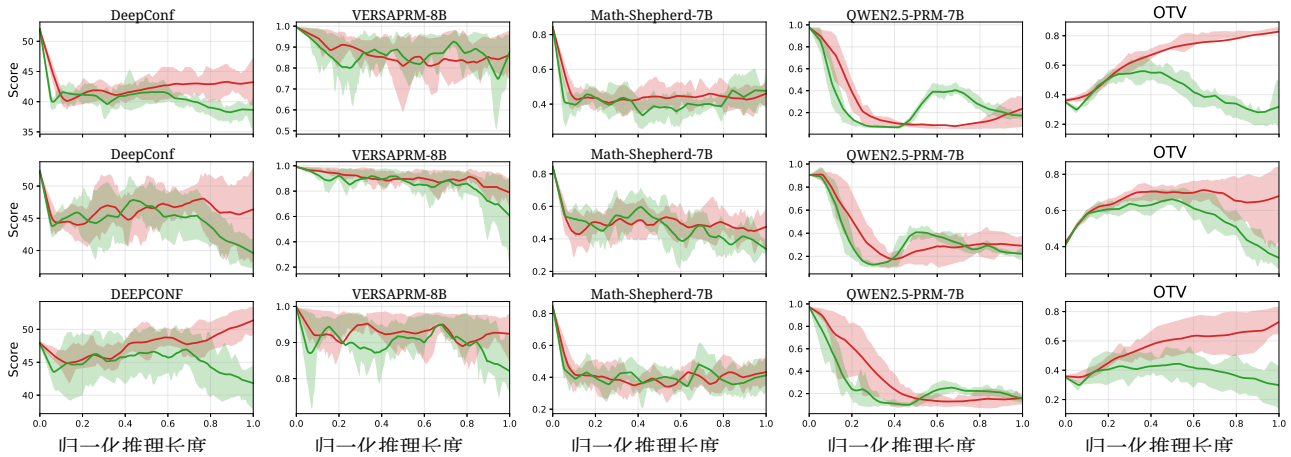


图 *2.* 三个具有代表性的 AIME24 问题（即，#3、#9 和 #22）上的置信度动态。对于每个预测器，我们绘制了 32 条采样推理轨迹的平均置信度轨迹，并分别展示以正确（红色）和错误（绿色）最终答案结束的轨迹。阴影区域表示各轨迹间的分位数范围，用于概括轨迹间的变异性。

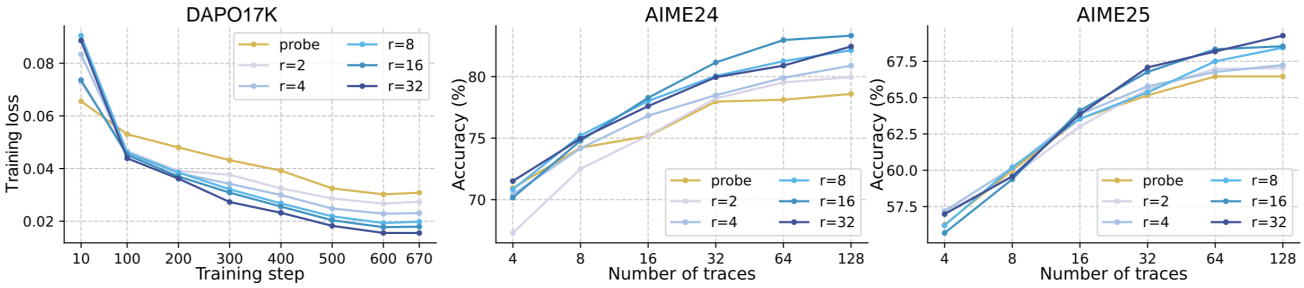


图 *3.* 验证器容量（即，LoRA 秩）对训练动态和下游投票准确率的影响。左：“probe”基线（仅训练回归头，不使用 LoRA，也不使用键值缓存）和不同 LoRA 秩下的 OTV 在优化步数增加时的验证器训练损失。中/右：AIME 上的加权多数投票准确率随采样轨迹数量的变化。所有结果均为 64 次运行的平均值。

策略中，Halve@300在准确率与效率之间提供了最有利的权衡（另见附录 B.4），其中 OTV 展现出明显优势。总体而言，这些结果表明，模型专属验证为通用的基于分数的选择提供了一种可靠且计算高效的替代方案。

我们还在预训练基础模型和多种架构（例如，LLaMA）上进一步验证了 OTV 的泛化能力，并观察到一致的改进。详情见附录B.2 和B.3。

4.3. 可视化

图 2定性展示了在三个具有代表性的 AIME 24 问题（即，第 3、9 和 22 题）上，不同置信度预测器在生成过程中的演变方式。对于每个验证器，我们分别针对正确（红色）和错误（绿色）解答，绘制了 *32* 条采样轨迹上的平均 置信度轨迹，并分别展示正确和错误解答。每条平均曲线周围的阴影区域反映了跨轨迹变异性（即，0.2–0.8 的 分位数间距），从而便于比较不同验证器的总体趋势。其他 OTV 可视化结果

所有AIME问题的tions见附录F。

在全部三个案例中都呈现出一致的模式：Deep-Conf 和以往的过程奖励模型通常会产生高度交织的置信度曲线，正确轨迹与错误轨迹之间存在大量重叠，这限制了它们在解码早期区分有希望的候选项与失败候选项的能力。相比之下，OTV 展现出明显更清晰的分层：对于正确轨迹，置信度通常会随着推理过程的推进而上升，而错误轨迹的置信度则相对受到抑制。这种增强的可分性与第 4.2 节所报告的定量提升相一致，也有助于解释 OTV 在基于得分的选择、剪枝和提前终止机制下更强的性能。

附录 G 提供了更多词元级可视化，将置信度曲线与生成文本对齐。这种更细粒度的分析展示了 OTV 的分数如何在不同解答中变化——通常会在关键计算步骤之后急剧上升——从而提升了单个推理步骤粒度上的可解释性。

表 3. 用于训练 OTV 的词元级伪置信度标注启发式方法消融实验。

	MAJ@128		BEST-OF-N		HALVE@300	
	AIME24	AIME25	AIME24	AIME25	AIME24	AIME25
恒定标签	81.41±1.65	68.49±2.04	72.76 (4,340)	54.27 (4,362)	66.41 (3,951)	**49.27** (4,503)
S形斜坡	80.05±1.91	66.51±0.92	**77.40** (4,617)	**56.41** (4,426)	68.44 (4,210)	46.30 (4,294)
噪声扰动斜坡	81.87±1.65	67.92±1.61	72.60 (5,226)	54.90 (5,152)	**71.30** (4,134)	48.33 (5,736)
阶梯式斜坡	82.40±1.50	68.80±1.60	77.29 (5,351)	55.36 (5,452)	70.63 (4,231)	37.60 (7,095)
线性渐变（默认）	**83.33**±1.57	**69.32**±1.46	73.44 (5,447)	53.91 (5,416)	67.03 (4,132)	49.02 (3,170)

表 4. 轨迹级置信度聚合与筛选对加权多数投票的影响。我们将 OTV 的令牌级置信度估计聚合于最后几个词元，从而转换为单个轨迹级分数；舍弃置信度最低的 ρ 部分推理轨迹（即，0%、25%、50% 和 75%），并评估三种聚合算子。

	max				mean				min			
	0%	25%	50%	75%	0%	25%	50%	75%	0%	25%	50%	75%
所有词元	75.78	76.82	80.42	80.36	75.94	79.53	82.19	80.00	80.10	80.83	81.25	79.17
最后10% 词元	76.77	80.21	82.76	81.46	82.19	83.02	83.28	82.19	83.07	83.07	83.12	82.29
最后 1 ，600 词元	75.78	78.65	81.25	82.14	80.99	82.71	83.23	82.24	83.23	83.23	82.97	79.95
最后400 词元	77.50	80.36	82.66	82.86	81.93	82.81	83.33	83.23	82.86	83.23	83.28	83.75
最后100 词元	78.39	80.21	82.34	81.82	82.08	82.97	83.33	82.76	82.86	82.60	82.76	**83.96**

4.4. 消融研究

我们开展消融研究，以量化各项设计选择对 OTV 性能的贡献。具体而言，我们考察：1）验证模块中使用的 LoRA 秩；2）替代伪置信度标注规则；3）轨迹级置信度聚合与筛选对加权多数投票的影响。

LoRA 秩的影响我们首先改变 LoRA 秩 r，同时保持底层推理器不变，并评估随着采样轨迹数量增加，加权多数投票下的性能。如图3所示，我们观察到明显的容量—性能权衡：增大 r可降低验证器的训练损失并提高下游准确率；即使秩处于中等水平，也已能获得显著提升（例如，$r = 16$）。重要的是，OTV 的表现始终优于“探针”基线；该基线仅训练回归头（即，不进行 LoRA 微调，也不复用键值缓存）。这表明，验证器既受益于 1）额外的适应能力，也受益于 2）通过键值缓存更充分地访问推理轨迹，而非仅依赖最终层隐藏状态（Zhang 等人，2025a）。

伪置信度标注 我们接下来消融启发式伪置信度目标 c_t，其定义见第3.3节，同时保持投票协议固定。表3 表明，常数标签在整条轨迹上均匀施加事后式目标，往往会让验证器偏向更短的轨迹，并略微降低加权多数投票的准确率。相比之下，基于递增函数的替代方案取得了相当的性能，同时通过施加朝向最终结果的单调递进，缓解了这种均匀事后偏差。因此，我们采用线性斜坡 作为默认方案，因其简单且性能稳定。

轨迹级置信度聚合与筛选 我们进一步研究如何将令牌级置信度估计映射为用于加权多数投票的单一轨迹级分数。遵循 DeepConf 式协议（Fu 等人，2025），我们在每条轨迹的末段上聚合置信度，然后移除最低的 ρ 比例的推理轨迹，再进行投票。表 4 揭示了两个一致的趋势。首先，对轨迹尾部进行聚合（例如，最后 100 个词元）比使用完整轨迹更为可靠，这支持了如下直觉：后期步骤验证能更好地捕捉已充分形成的推理状态和最终答案。其次，中等至较为激进的筛选通过抑制低置信度候选提升了鲁棒性；在我们的设定中，对大约最后100 个词元进行均值/最小值聚合的效果最佳。

5. 结论

在这项工作中，我们提出了 OTV，这是一种高效、针对特定模型的计算方法，用于估计令牌级推理正确性。OTV 为推理大语言模型配备了一条 LoRA 门控验证通路，只有插入专用的 `[ToT]` 时才会激活，使模型能够进入验证模式，同时不影响其默认推理行为。OTV 通过探测模型的键值缓存，并借助一个小型回归头生成标量置信度分数，以每个查询仅需一次前向传播的代价支持随时验证。针对多个推理大语言模型和数学基准测试开展的实验表明，与现有的内部和外部验证器相比，OTV 能够持续提升并行思考聚合和剪枝策略的效果，同时通过置信度引导的提前终止显著提高效率。我们还在附录 D 中讨论了局限性和未来的研究方向。

影响声明

本文旨在推动机器学习领域的发展。我们的工作可能带来许多社会影响，但我们认为这些影响都无需在此特别指出。

参考文献

Ankner, Z., Paul, M., Cui, B., Chang, J. D., and Ammanabrolu, P. 大声批评奖励模型。*arXiv*预印本 *arXiv:2408.11791*，2024。

Azaria, A. and Mitchell, T. 大语言模型的内部状态知道它何时在说谎。*arXiv*预印本 *arXiv:2304.13734*，2023年。

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., R´e, C., and Mirhoseini, A. 大型语言猴子：通过重复采样扩展推理计算。*arXiv*预印本 *arXiv:2407.21787*，2024。

Burns, C., Ye, H., Klein, D. 和 Steinhardt, J. 无需监督即可发现语言模型中的潜在知识。*arXiv* 预印本 *arXiv:2212.03827*，2022年。

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. O., Kaplan, J., Edwards, H., Burda, Y.,Joseph, N., Brockman, G. 等人。评估基于代码训练的大语言模型。*arXiv* 预印本 *arXiv:2107.03374*，2021。

Chen, X., Xu, J., Liang, T., He, Z., Pang, J., Yu, D., Song, L., Liu, Q., Zhou, M., Zhang, Z. 等人。不要想那么多， 2+3=? 关于 o1 类大语言模型的过度思考。*arXiv* 预印本 *arXiv:2412.21187*，2024a。

Chen, Z., White, M., Mooney, R., Payani, A., Su,Y., and Sun, H. 何时树搜索对大语言模型规划有用？这取决于判别器。*arXiv* 预印本 *arXiv:2402.10890*，2024b。

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., 等。训练验证器以解决数学应用题。*arXiv* 预印本 *arXiv:2110.14168*，2021。

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., 等。Gemini 2.5：通过先进的推理、多模态、长上下文和下一代智能体能力，推动前沿发展。*arXiv* 预印本 *arXiv:2507.06261*，2025。

Fadeeva, E., Rubashevskii, A., Shelmanov, A., Petrakov, S., Li, H., Mubarak, H., Tsymbalov, E., Kuzmin, G., Panchenko, A., Baldwin, T. 等。通过词元级不确定性量化对大语言模型的输出进行事实核查。*arXiv* 预印本 *arXiv:2403.04696*，2024。

Feng, S., Kong, X., Ma, S., Zhang, A., Yin, D., Wang, C., Pang, R. 和 Yang, Y. 通过扭曲序贯蒙特卡洛对数学问题进行逐步推理。*arXiv* 预印本 *arXiv:2410.01920*，2024。Fu, Y., Wang, X., Tian, Y. 和 Zhao, J. 自信深度思考。*arXiv* 预印本 *arXiv:2508.15260*，2025。Ghosal, S. S., Chakraborty, S., Reddy, A., Lu, Y., Wang, M., Manocha, D., Huang, F., Ghavamzadeh, M. 和 Bedi, A. S. 思考得更多总是有帮助吗？理解推理模型中的测试时缩放。*arXiv* 预印本 *arXiv:2506.04210*，2025。

Golovneva, O., Chen, M., Poff, S., Corredor, M., Zettlemoyer, L., Fazel-Zarandi, M., and Celikyilmaz, A. ROSCOE：一套用于逐步推理评分的指标。*arXiv*预印本 *arXiv:2212.07919*，2022年。

Guan, X., Zhang, L. L., Liu, Y., Shang, N., Sun, Y., Zhu, Y., Yang, F., and Yang, M. rStar-Math：小型大语言模型能够通过自进化深度思考掌握数学推理。*arXiv*预印本 *arXiv:2501.04519*，2025。

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., 等。DeepSeek-R1：通过强化学习激励大语言模型的推理能力。*arXiv*预印本 *arXiv:2501.12948*，2025。

Hosseini, A., Yuan, X., Malkin, N., Courville, A., Sordoni, A., 和 Agarwal, R. V-STaR：为自我教学的推理器训练验证器。发表于语言建模会议，2024。

Hsu, C.-J., Buffelli, D., McGowan, J., Liao, F.-T., Chen, Y.-C., Vakili, S., 和 Shiu, D.-s. 群体思考：多个并发推理智能体以词元级粒度协作。*arXiv*预印本 *arXiv:2505.11107*，2025。

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., 和 Chen, W. LoRA：大语言模型的低秩自适应。发表于国际学习表征会议，2022年。

Huang, C., Huang, L., Leng, J., Liu, J., 和 Huang, J. 通过自校准实现高效的测试时缩放。*arXiv*预印本 *arXiv:2503.00031*，2025。

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X. 和 Zhou, D. 大型语言模型尚无法自我纠正推理。*arXiv* 预印本 *arXiv:2310.01798*，2023年。

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Car- ney, A., 等。OpenAI o1系统卡。*arXiv* 预印本 *arXiv:2412.16720*，2024。

Kang, Z., Zhao, X. 和 Song, D. 通过自确定性实现大语言模型的可扩展最佳-N 选择。*arXiv* 预印本 *arXiv:2502.18581*，2025。

Lee, J. 和 Hockenmaier, J. 逐步推理轨迹评估：综述。*arXiv* 预印本 *arXiv:2502.12289*，2025。

Lee, J. H.、Yang, J. Y.、Heo, B.、Han, D.、Kim, K.、Yang, E. 和 Yoo, K. M. 词元监督价值模型：增强大语言模型的数学问题求解能力。载于国际学习表征会议，2025。

Lewis, P.、Perez, E.、Piktus, A.、Petroni, F.、Karpukhin, V.、Goyal, N.、K¨uttler, H.、Lewis, M.、Yih, W.-t.、Rockt¨aschel, T. 等。面向知识密集型 NAIME2024LP 任务的检索增强生成。载于神经信息处理系统进展，第 9459–9474 页，2020。

Li, P., Zhou, Y., Muhtar, D., Yin, L., Yan, S., Shen, L., Liang, Y., Vosoughi, S., 和 Liu, S. 扩散语言模型在解码前就知道答案。*arXiv* 预印本 *arXiv:2508.19982*，2025。

Lifshitz, S., McIlraith, S. A., 和 Du, Y. 多智能体验证：借助多个验证器扩展测试时计算。*arXiv* 预印本 *arXiv:2502.20379*，2025。

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., 和 Cobbe, K. 让我们逐步验证。发表于国际学习表征会议，2024。

Lin, S., Hilton, J., 和 Evans, O. 教会模型用文字表达自身的不确定性。*arXiv* 预印本 *arXiv:2205.14334*，2022年。

Liu, Z., Chen, Y., Shoeybi, M., Catanzaro, B., 和 Ping, W. AceMath：通过后训练和奖励建模推进前沿数学推理。*arXiv* 预印本 *arXiv:2412.15084*，2024。

Lu, J., Dou, Z., Wang, H., Cao, Z., Dai, J., Feng, Y., 和 Guo, Z. AutoPSV：自动化过程监督验证器。载于《神经信息处理系统进展》，第 79935–79962 页，2024。

Luo, L., Liu, Y., Liu, R., Phatale, S., Guo, M., Lara, H., Li, Y., Shu, L., Zhu, Y., Meng, L. 等。通过自动化过程监督改进语言模型的数学推理。*arXiv* 预印本 *arXiv:2406.06592*，2024。

MAA. 2024年美国数学邀请赛（AIME）。竞赛题目与解答，2024年。网址 https://www.maa.org/math-competitions/aime。

MAA. 2025美国数学邀请赛（AIME）。竞赛题目与解答，2025年。网址 https://www.maa.org/math-competitions/aime。Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Cand`es, E., and Hashimoto, T. s1：简单的测试时缩放。*arXiv*预印本 *arXiv:2501.19393*，2025年。Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., 等。通过人类反馈训练语言模型以遵循指令。载于神经信息处理系统进展，第 27730–27744页，2022年。

Samragh, M., Kundu, A., Harrison, D., Nishu, K., Naik, D., Cho, M., and Farajtabar, M. 你的大语言模型知道未来：揭示其多词元预测潜力。*arXiv* 预印本 *arXiv:2507.11851*，2025年。

Schick, T., Dwivedi-Yu, J., Dess`ı, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer：语言模型可以自学使用工具。载于神经信息处理系统进展，第 68539–68551 页，2023年。

Setlur, A., Nagpal, C., Fisch, A., Geng, X., Eisenstein, J., Agarwal, R., Agarwal, A., Berant, J., and Kumar, A. Rewardingprogress：扩展用于大语言模型推理的自动化过程验证器。*arXiv* 预印本 *arXiv:2410.08146*，2024。

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., 等。DeepSeekMath：推动开放语言模型数学推理的极限。*arXiv* 预印本 *arXiv:2402.03300*，2024。

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., 和 Christiano, P. F. 通过人类反馈学习摘要。在神经信息处理系统进展，第 3008–3021 页，2020。

Suzgun, M., Scales, N., Sch¨arli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., 等。具有挑战性的 BIG-Bench 任务，以及思维链能否解决这些任务。在计算语言学协会论文集，第 13003–13051 页，2023年。

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., 等。Kimi k1.5：利用大语言模型扩展强化学习规模。*arXiv* 预印本 *arXiv:2501.12599*，2025。

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., 和 Higgins, I. 使用基于过程和结果的反馈解决数学应用题。*arXiv* 预印本 *arXiv:2211.14275*，2022年。

Vaswani, A.、Shazeer, N.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, A. N.、Kaiser, Ł. 和 Polosukhin, I. 注意力就是你所需要的一切。载于神经信息处理系统进展，2017年。

Venktesh, V.、Rathee, M. 和 Anand, A. 信任，但要验证！一篇关于测试时缩放验证设计的综述。*arXiv*预印本 *arXiv:2508.16665*，2025年。

Wang, H.、Xiong, W.、Xie, T.、Zhao, H. 和 Zhang, T. 通过多目标奖励建模和专家混合实现可解释的偏好。*arXiv*预印本*arXiv:2406.12845*，2024年。

Wang, P.、Li, L.、Shao, Z.、Xu, R.、Dai, D.、Li, Y.、Chen, D.、Wu, Y. 和 Sui, Z. Math-Shepherd：无需人工标注，逐步验证并强化大语言模型。*arXiv*预印本 *arXiv:2312.08935*，2023a。

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., 和 Zhou, D. 自一致性提升了语言模型中的思维链推理效果。载于国际学习表征会议，2023b。

Wang, Y., Zhang, P., Huang, S., Yang, B., Zhang, Z., Huang, F., 和 Wang, R. 采样高效的测试时扩展：在早期解码中自估计最佳的-*N* 采样。*arXiv* 预印本 *arXiv:2503.01422*，2025。

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E. H., Le, Q. V., 和 Zhou, D. 思维链提示能够激发大语言模型的推理能力。载于神经信息处理系统进展，第 24824–24837 页，2022年。

Wen, H., Su, Y., Zhang, F., Liu, Y., Liu, Y., Zhang, Y.-Q., a nd Li, Y. ParaThinker：将原生并行思考作为扩展大语言模型测试时计算的新范式。*arXiv* 预印本 *arXiv:2509.04475*，2025。

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. 大语言模型能否表达其不确定性？对大语言模型置信度引出的实证评估。发表于国际学习表征会议，2024。

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., 等。Qwen2.5-Math 技术报告：通过自我改进迈向数学专家模型。*arXiv* 预印本 *arXiv:2409.12122*，2024。

Yang, A.、Li, A.、Yang, B.、Zhang, B.、Hui, B.、Zheng, B.、Yu, B.、Gao, C.、Huang, C.、Lv, C. 等。Qwen3 技术报告。*arXiv* 预印本 *arXiv:2505.09388*，2025a。

Yang, X.、An, Y.、Liu, H.、Chen, T. 与 Chen, B.。Multiverse：你的语言模型会秘密决定如何并行化并合并生成。*arXiv* 预印本 *arXiv:2506.09991*，2025b。

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. 思维树：使用大语言模型进行审慎的问题求解。载于神经信息处理系统进展，第 11809–11822 页，2023年。

Ye, Z., Melo, L. C., Kaddar, Y., Blunsom, P., Staton, S., and Gal, Y. 使用生成式奖励模型进行不确定性感知的逐步验证。*arXiv* 预印本 *arXiv:2502.11250*，2025。

Yu, F., Gao, A., and Wang, B. OVM：用于数学推理规划的结果监督价值模型。*arXiv* 预印本 *arXiv:2311.09724*，2023年。

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. MetaMath：为大语言模型自举生成数学问题。载于国际学习表征会议，2024。

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L. 等。DAPO：大规模开源大语言模型强化学习系统。*arXivpreprint arXiv:2503.14476*，2025。

Zeng, T., Zhang, S., Wu, S., Classen, C., Chae, D., Ewer, E., Lee, M., Kim, H., Kang, W., Kunde, J. 等。VersaPRM：基于合成推理数据的多领域过程奖励模型。*arXivpreprint arXiv:2502.06737*，2025。

Zhang, A., Chen, Y., Pan, J., Zhao, C., Panda, A., Li, J. 和 He, H.。推理模型知道自己何时正确：探测隐藏状态以进行自我验证。*arXiv preprintarXiv:2504.05419*，2025a。

Zhang, D.、Huang, X.、Zhou, D.、Li, Y. 和 Ouyang, W. 通过使用 LLaMa-3 8B 进行蒙特卡洛树自我改进，获取 GPT-4 级别的数学奥林匹克竞赛解答。*arXiv* 预印本 *arXiv:2406.07394*，2024a。

Zhang, D.、Zhoubian, S.、Hu, Z.、Yue, Y.、Dong, Y. 和 Tang, J. ReST-MCTS*：通过过程奖励引导的树搜索进行大语言模型自训练。发表于神经信息处理系统进展，第 64735–64772 页，2024b。

Zhang, L.、Hosseini, A.、Bansal, H.、Kazemi, M.、Ku- mar, A. 和 Agarwal, R. 生成式验证器：将奖励建模为下一词元预测。*arXiv* 预印本 *arXiv:2408.15240*，2024c。

Zhang, Y., Emma, S. Y., En, A. L. J., and Dong, J. S. RvLLM：具备领域知识的 LLM 运行时验证。*arXiv* 预印本 *arXiv:2505.18585*，2025b。

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. 数学推理过程中开发过程奖励模型的经验教训。*arXiv* 预印本 *arXiv:2501.07301*，2025c。Zhao, W., Aggarwal, P., Saha, S., Celikyilmaz, A., Weston, J., and Kulikov, I. 多数意见并非总是正确：用于解答聚合的强化学习训练。*arXiv* 预印本 *arXiv:2509.06870*，2025。Zheng, C., Zhang, Z., Zhang, B., Lin, R., Lu, K., Yu, B., Liu, D., Zhou, J., and Lin, J. ProcessBench：识别数学推理中的过程错误。发表于计算语言学协会年会，第 1009–1024 页，2025a。Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. 使用 MT-Bench 和 Chatbot Arena 评判 LLM-as-a-Judge。发表于神经信息处理系统进展，第 46595–46623 页，2023。Zheng, T., Zhang, H., Yu, W., Wang, X., Dai, R., Liu, R., Bao, H., Huang, C., Huang, H., and Yu, D. Parallel-R1：通过强化学习实现并行思考。*arXiv* 预印本 *arXiv:2509.07980*，2025b。Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y. LlamaFactory：统一且高效的100+ 语言模型微调。*arXiv* 预印本 *arXiv:2403.13372*，2024。Zhong, J., Li, Z., Xu, Z., Wen, X., Li, K., and Xu, Q. Solve-Detect-Verify：借助灵活的生成式验证器实现推理时扩展。*arXiv* 预印本 *arXiv:2505.11966*，2025。

A. 算法描述

算法1 概述了并行化的 OTV 训练过程：对于每条采样的推理轨迹，我们 1）缓存冻结的推理器产生的 KV 状态，2）构造令牌级伪置信度目标，3）通过在所有探测位置插入真实令牌 [ToT]，执行单次并行验证过程。算法 2 描述了 OTV 推理过程：单个验证令牌 [ToT] 探测任意缓存前缀，并返回令牌级正确性估计。我们用浅紫色底纹表示推理模式步骤（即，轨迹生成和 KV 缓存），用浅黄色底纹表示验证模式步骤（即，[ToT]探测）。

算法 1 并行化 OTV 训练

要求： 数据集 $\mathcal{D}$

1: **轨迹采样 + KV 缓存**。对于每个 $(\boldsymbol{q},\ a) \in \mathcal{D}$，采样一条推理轨迹 $\boldsymbol{x}_{1:T} \sim \text{LLM}(\boldsymbol{q})$ 并缓存每层 KV 状态 $\mathcal{C}_T$，其中 $\mathcal{C}_t = \{\mathbf{K}_t^{(l)}、\mathbf{V}_t^{(l)}\}_{l=1}^{L}$

2: **伪置信度目标**。根据轨迹级结果标签 $y \in \{0,1\}$ 计算词元级目标 $\{c_t\}_{t=1}^{T}$，其中 y 是通过将模型的最终答案与真实标注 a 进行比较得出的

3: **单次并行探测**。构造探测序列 $[\text{ToT}]_{1:T+1}$，并应用三角探测掩码 $\mathbf{M}$，使得探测 $t+1$ 仅关注缓存前缀 $1:t$；在一次前向传播中计算所有词元级预测 $\{\hat{c}_t\}$

4: **优化**。最小化响应令牌上的均方误差，并更新 LoRA 和回归头参数

该实现等价于发出 $T+1$ 个独立的单令牌验证查询，但会在单次前向传播中执行这些查询，同时复用相同的缓存的前缀状态。

算法 2 OTV 推理（单令牌探测）

要求： 提示/问题 $\boldsymbol{q}$，部分轨迹前缀 $\boldsymbol{x}_{1:t}$，缓存的 KV 状态 $\mathcal{C}_t$

1: **前缀生成 + 缓存**。 生成（或复用）部分轨迹 $\boldsymbol{x}_{1:t}$ 和键值缓存 $\mathcal{C}_t$

2: **单令牌验证查询**。 在位置 $t+1$ 插入单个 [ToT]，复用 $\mathcal{C}_t$，并运行前向传播于 LoRA 门控验证器之下，以获得 $\hat{c}_t \in [0,1]$

3: **轨迹级评分（可选）**。将 $\{\hat{c}_t\}$（例如，在最终片段上）聚合为单个分数，用于排序和剪枝

B. 其他实验细节

B.1. 替代伪置信度标注规则

OTV 使用结果监督 $y \in \{0,1\}$ 来定义稠密词元级目标 $c_t \in [0,1]$。除了默认的线性斜坡外，我们还评估了四种替代方案。

- 恒定标签采用一种事后式目标，假设模型在整个轨迹中都“知道”最终结果：

$$c_t = y. \tag{12}$$

- *S*形斜坡提供 是一种平滑的参数化替代方案，可在渐进过渡和突变过渡之间进行插值 s:

$$c_t = \text{sigmoid}\left((2y-1)\alpha\frac{t}{T}\right), \tag{13}$$

其中 $\alpha > 0$ 控制形状：较小的 α 会产生近线性斜坡，而较大的 α 则会产生向最终标签的更急剧过渡，趋近于阶跃式变化（即，恒定标签）。我们默认设置 $\alpha = 4$。

表5。 使用*MetaMath-LLaMA-7B* 和*MetaMath-Mistral-7B* 作为骨干模型，在GSM8K上得到的加权多数投票准确率。结果报告为64次运行的均值 ± 标准差。

	N	1	4	8	16	32	64
LLaMA-7B	PASS@k	61.87±0.74	75.01±0.79	81.07±0.49	85.36±0.48	88.68±0.35	91.27±0.38
	MAJ@k	61.87±0.74	67.29±0.88	69.57±0.43	70.53±0.44	71.47±0.30	71.79±0.30
	DEEPCONF	61.87±0.74	68.64±0.80	70.63±0.60	71.76±0.58	72.06±0.20	72.32±0.21
	OTV（$r = 2$）	61.87±0.74	**71.13**±0.74	**73.50**±0.47	**74.55**±0.52	**74.80**±0.37	**75.16**±0.22
	OTV（$r = 16$）	61.87±0.74	70.72±0.81	72.82±0.35	73.80±0.32	74.37±0.28	74.83±0.29
Mistral-7B	PASS@k	66.02±1.18	83.16±0.65	88.46±0.50	91.58±0.44	93.99±0.28	95.87±0.24
	MAJ@k	66.02±1.18	72.60±0.49	77.33±0.30	79.04±0.37	80.21±0.40	80.56±0.33
	DEEPCONF	66.02±1.18	75.63±0.90	78.62±0.83	79.61±0.34	80.81±0.37	81.31±0.13
	OTV ($r = 2$)	66.02±1.18	**79.52**±0.73	82.05±0.44	83.12±0.25	84.06±0.24	84.24±0.30
	OTV ($r = 16$)	66.02±1.18	79.51±0.57	**82.27**±0.30	**83.54**±0.42	**84.18**±0.31	**84.48**±0.21

• 噪声扰动斜坡。为了在保持全局单调趋势的同时对不确定性中的局部波动进行建模，我们用加性正弦项和高斯噪声扰动线性基线：

$$c_t = \text{clip}\left(\text{linear}(t) + \beta \sin\left(\frac{2\pi\omega t}{T}\right) + \epsilon,\ 0, 1\right), \tag{14}$$

其中 β为振幅， ω为频率， $\epsilon \sim \mathcal{N}(0,\ \sigma^2)$。我们默认设置 $\beta = 0.1$ 和 $\omega = 3$ ult.

• 阶梯式斜坡（按推理步骤划分）。有些推理轨迹天然由离散的推理步骤组成。我们使用“\n\n”分隔符将轨迹划分为 Γ 个步骤，并令 $\gamma(t)$ 表示词元 t 所属推理步骤的索引。然后定义

$$c_t = 0.5 + (y - 0.5)\frac{\gamma(t)}{\Gamma}, \tag{15}$$

由此得到一条分段常数轨迹，仅在新步骤开始时递增。

B.2. 预训练基础模型上的评估

为进一步探究观察到的提升是否仅限于后训练推理模型，我们还在预训练基础模型 *Qwen3-4B-Base*上评估 OTV。在此设置下，我们在 MetaMathQA（Yu 等人，2024）上训练，并在 GSM8K（Cobbe 等人，2021）上评估，以衡量原始数学推理能力的提升。如图 4所示，OTV 的表现显著优于 Maj@k，并缩小了与 Pass@k 之间的差距，这表明即使没有指令微调，验证信号仍然有效。

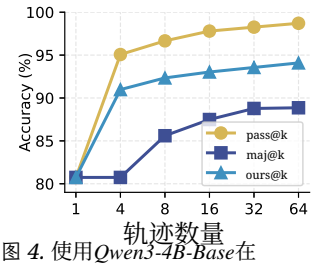


图 4. 使用*Qwen3-4B-Base*在 GSM8K 上进行评估。

B.3. 在其他模型系列上的评估

为评估 OTV 在 *Qwen* 系列之外的架构通用性，我们将 OTV 应用于 *MetaMath-LLaMA-7B*[3] 和 *MetaMath-Mistral-7B*[4] 模型，在低资源微调设置下（仅使用 1 万条 MetaMathQA 子集数据）对 GSM8K 进行评估。如表 5所示，在不同样本量下，OTV 的表现始终优于多数投票和 DeepConf；提高 LoRA 秩通常也能提升性能。

B.4. 推理时验证开销

OTV 的验证调用次数取决于聚合策略（以 $N = 128$ 为例）。

• **Drop@10**（8，256 次调用）每生成 10 个词元，就剪除得分最低的一条推理轨迹，直到只剩下一条。在第一个检查点（$t = 10$），验证全部 128 条推理轨迹并丢弃得分最低的一条；在第二个检查点（$t = 20$），剩余的

[3]https://huggingface.co/meta-math/MetaMath-7B-V1.0[4]
https://huggingface.co/meta-math/MetaMath-Mistral-7B

127 条轨迹已验证 推理轨迹经过验证后再丢弃一条，以此类推。验证调用总数为等差数列之和：

$$\#\text{Calls} = \sum_{n=1}^{N} n = \frac{N(N+1)}{2} = \frac{128 \times 129}{2} = 8,256. \tag{16}$$

• **Stop@600** （128 次调用）是一种"仅验证一次"的策略，仅当轨迹达到第 600 个词元时才触发验证器（如果生成提前终止，则在更早时触发）。因此， N 条候选轨迹中的每一条都恰好被验证一次：

$$\#\text{Calls} = N = 128. \tag{17}$$

• **Halve@300** （254 次调用）每 300 个词元将候选池减半。也就是说，我们在 $t = 300$ 处验证 128 条轨迹，在 $t = 600$ 处验证 64 条，在 $t = 900$ 处验证 32 条，依此类推。调用总数遵循几何级数，其上界为：

$$\#\text{Calls} \approx \sum_{k=0}^{\lceil \log_2(N-1) \rceil} \left\lceil \frac{N}{2^k} \right\rceil = 128 + 64 + 32 + \cdots + 4 + 2 = 254. \tag{18}$$

C. 理论分析

本节将验证视为对轨迹前缀 $\boldsymbol{s}_t = [\boldsymbol{q}, \boldsymbol{x}_{1:t}]$（等价地，其键值缓存）进行平方损失回归，从而形式化 OTV 在不同伪置信度标注规则下所学习的内容。

C.1. 设定

设基础推理器诱导出自回归策略 $\pi_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{q},\ \boldsymbol{x}_{1:t})$，并采样得到终止正确性为 $y \in \{0, 1\}$ 的推理轨迹 $\boldsymbol{x}_{1:T} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$。对于任意伪置信度标注规则 $c_t = c(t,\ T,\ y)$，OTV 可最小化逐令牌均方误差：

$$\ell(\boldsymbol{\phi}) = \mathbb{E}\Big[\big(f_{\boldsymbol{\phi}}(\boldsymbol{s}_t) - c(t, T, y)\big)^2\Big], \tag{19}$$

其中， $f_{\boldsymbol{\phi}}(\cdot)$ 表示验证器，其由向量 $\boldsymbol{\phi}$ 参数化；期望是对从基础推理器采样得到的轨迹 $\boldsymbol{x}_{1:T} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{q})$ 取值。

C.2. 均方误差下的最优预测器

命题 C.1 (均方误差下的风险最小化器)。 对于任意固定的 t 和任意状态 $\boldsymbol{s}_t$，在所有可测函数 $f_{\boldsymbol{\phi}}(\cdot)$ 中，条件风险 $\mathbb{E}\big[(f_{\boldsymbol{\phi}}(\boldsymbol{s}_t) - c(t, T, y))^2 \mid \boldsymbol{s}_t\big]$ 的最小化器是条件期望

$$f_{\boldsymbol{\phi}^\star}(\boldsymbol{s}_t) \ = \ \mathbb{E}[c(t, T, y) \mid \boldsymbol{s}_t]\,. \tag{20}$$

证明。 固定 $\boldsymbol{s}_t$，并考虑任意标量 a。根据偏差-方差分解，

$$\mathbb{E}\big[(a - c(t, T, y))^2 \mid \boldsymbol{s}_t\big] = \big(a - \mathbb{E}[c(t, T, y) \mid \boldsymbol{s}_t]\big)^2 + \mathrm{Var}\left(c(t, T, y) \mid \boldsymbol{s}_t\right), \tag{21}$$

其在 $a = \mathbb{E}[c(t,\ T,\ y) \mid \boldsymbol{s}_t]$ 处取最小值。 □

这表明，伪置信度标注规则 c_t 决定了 OTV 在训练期间试图近似的条件统计量。

C.3. 恒定标签可恢复蒙特卡洛正确性值

对于恒定标签 规则 $c_{\text{const}}(t,\ T,\ y) = y$，最优预测器等于

$$V_{\text{MC}}(\boldsymbol{s}_t) \ := \ \mathbb{P}(y = 1 \mid \boldsymbol{s}_t) \ = \ \mathbb{E}[y \mid \boldsymbol{s}_t], \tag{22}$$

即，从 $\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{s}_t)$采样到的续写能够得出正确的最终答案的概率。

C.4. 线性斜坡引发逆长度偏好

考虑满足以下条件的线性斜坡 $c_{\text{lin}}(t,\ T,\ y) = 0.5 + (y - 0.5)\,\frac{t}{T}$：对于 $t \le T$，$c_{\text{lin}}(t,\ T,\ y) \in [0, 1]$，且 $c_{\text{lin}}(T,\ T,\ y) = y$。将来自 $\boldsymbol{s}_t$ 的正确补全和错误补全的期望逆长度定义为

$$\mu_+(\boldsymbol{s}_t) := \mathbb{E}\left[\frac{1}{T}\,\middle|\,\boldsymbol{s}_t,\, y = 1\right], \qquad \mu_-(\boldsymbol{s}_t) := \mathbb{E}\left[\frac{1}{T}\,\middle|\,\boldsymbol{s}_t,\, y = 0\right]. \tag{23}$$

命题 C.2. 在 c_{lin} 下的均方误差最优预测器为

$$f_{\phi^\star}(\boldsymbol{s}_t) = \underbrace{0.5 - \frac{t}{2}\mu_-(\boldsymbol{s}_t)}_{\text{baseline}} + V_{\text{MC}}(\boldsymbol{s}_t) \cdot \underbrace{\frac{t}{2}\big(\mu_+(\boldsymbol{s}_t) + \mu_-(\boldsymbol{s}_t)\big)}_{\text{gain}>0}, \tag{24}$$

其中 $V_{\text{MC}}(\boldsymbol{s}_t) = \mathbb{E}[y \mid \boldsymbol{s}_t]$。特别地，对于固定的 $\boldsymbol{s}_t$ 和 t，$f_{\phi^\star}(\boldsymbol{s}_t)$ 关于 $V_{\text{MC}}(\boldsymbol{s}_t$ 严格递增).

证明。 根据命题 C.1

$$f_{\phi^\star}(\boldsymbol{s}_t) = \mathbb{E}\left[0.5 + (y - 0.5)\frac{t}{T}\,\middle|\,\boldsymbol{s}_t\right] = 0.5 + t \cdot \mathbb{E}\left[\frac{y - 0.5}{T}\,\middle|\,\boldsymbol{s}_t\right]. \tag{25}$$

以 $y \in \{0, 1\}$ 为条件可得

$$\begin{aligned}\mathbb{E}\left[\frac{y - 0.5}{T}\,\middle|\,\boldsymbol{s}_t\right] &= \mathbb{P}(y = 1 \mid \boldsymbol{s}_t) \cdot \mathbb{E}\left[\frac{0.5}{T}\,\middle|\,\boldsymbol{s}_t, y = 1\right] + \mathbb{P}(y = 0 \mid \boldsymbol{s}_t) \cdot \mathbb{E}\left[\frac{-0.5}{T}\,\middle|\,\boldsymbol{s}_t, y = 0\right] \\ &= \frac{1}{2}V_{\text{MC}}(\boldsymbol{s}_t)\mu_+(\boldsymbol{s}_t) - \frac{1}{2}(1 - V_{\text{MC}}(\boldsymbol{s}_t))\mu_-(\boldsymbol{s}_t),\end{aligned} \tag{26}$$

并代入式（25）可得式（24）。由于 $\mu_+,\ \mu_- > 0,\ V_{\text{MC}}$ 的系数 $\frac{t}{2}\ (\mu_+ + \mu_-)\ > 0$，这意味着严格单调性。 □

式（24）表明，除了随 V_{MC}单调变化外，得分还受 μ_+ 和 μ_-的调节；固定 V_{MC} 时，较大的 μ_+（即，从预期来看，正确补全更短）会提高 $f_{\phi^\star}$，从而偏好更短的正确轨迹。

D. 未来工作

基于这些观察，有几个方向值得探索。首先是加强模型与验证器协同适应。我们可以不再将验证器视为固定的附加组件，而是探索联合训练或持续训练，使验证器跟踪基础模型的变化以及不断演变的解码分布。这自然将以模型为中心的改进（例如，更强的推理骨干网络、更好的校准和解码时优化）与以数据为中心的改进联系起来：OTV 的评分可用于筛选更高质量的训练轨迹、执行困难负样本挖掘，并通过优先标注或投入额外计算资源处理“高不确定性”案例来支持主动学习。

其次，改进词元级伪置信度标注是一个重要的机遇。当前方法根据最终结果生成稠密目标，但更丰富的监督信号可能会显著提升性能。一个方向是采用不确定性感知伪标签，表示部分进展、可逆错误或后期步骤失误，而不是强制置信度轨迹始终单调。另一个方向是对词元级监督进行混合自举：在结果标签的基础上，加入自一致性／集成分歧等辅助信号、步骤边界先验（例如，推理步骤之间的转换标记），并针对少量样本，从能力更强但成本更高的过程验证器中进行选择性蒸馏。这一过程可以迭代进行：每个改进后的验证器都会为后续训练轮次生成更高保真的伪标签，从而减少弱初始目标带来的系统性偏差。

第三，扩展验证器输出空间及其在决策中的作用十分必要，以便解锁更广泛的应用。例如，采用选择性预测的形式，例如从二元置信度转向“正确/错误/未知”三元信号，就能在模型不确定时支持弃权和风险控制路由。最后，值得在数学基准测试之外检验 OTV，包括代码推理和工具增强任务，并研究验证器引导的计算分配如何与长上下文设置交互（例如多步规划和多智能体/集成解码）。

E. 对解答扰动的敏感性

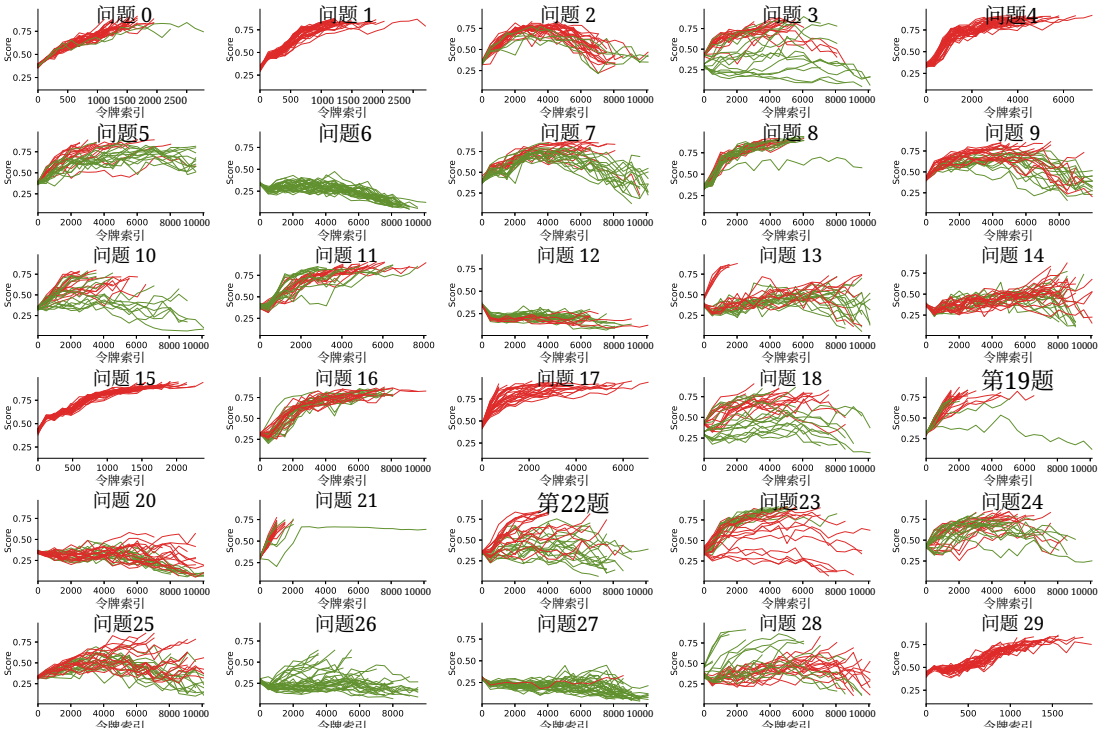
为检验其对表面修改与语义错误的鲁棒性，我们对同一段解答文本进行受控扰动（见表 6）。我们将保留语义的编辑（例如缩短/扩写、格式调整或替换为真实标注答案）与破坏逻辑的编辑（例如更改数字、改变逻辑或破坏最后一步）进行对比。经验结果表明，对于正确变体，OTV 的置信度会稳定上升；对于错误变体，置信度则会下降。这说明 OTV 反映的是内部推理的一致性，而非表面形式。

表6. 敏感性分析中使用的代表性扰动解答变体。每一行都对同一个基础解答进行受控编辑，以评估验证器的置信度对保留语义与破坏逻辑的修改有何反应。

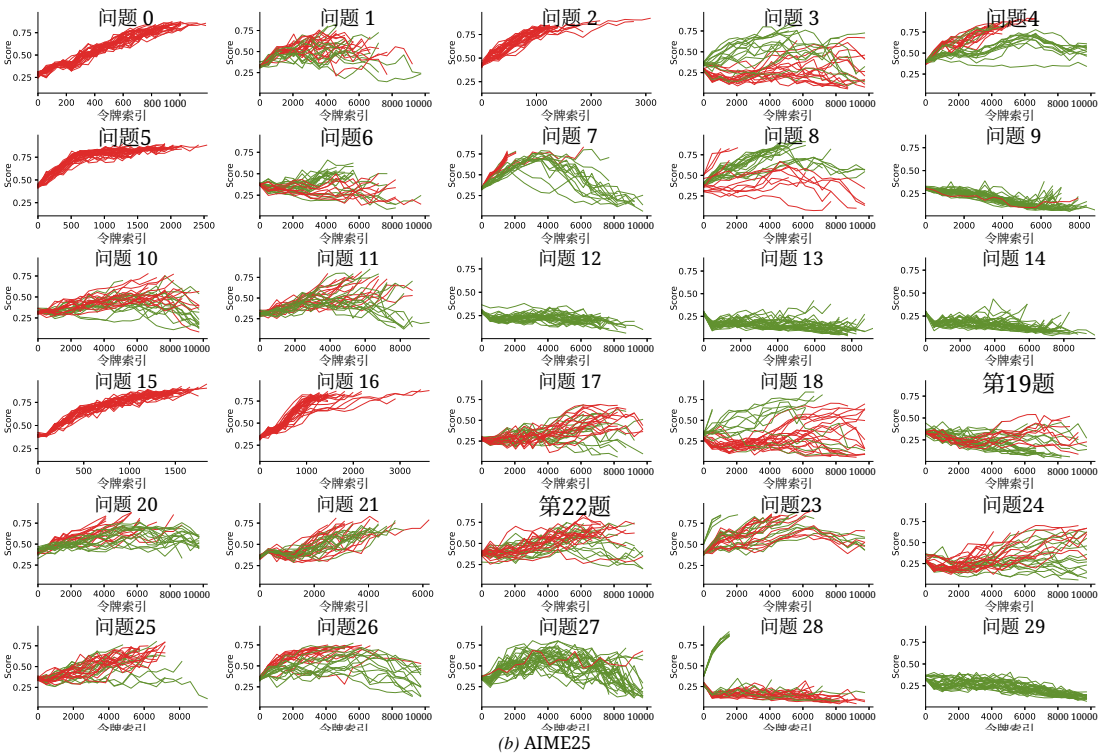
Type	Text
	保持语义的修改
真实标注	需要 $2/2 = 1$ 捆白色纤维\n因此织物总量为 $2 + 1 = 3$ 卷织物\n#### 3
原始	要解决这个问题，我们需要计算这件长袍所需的卷数，包括蓝色纤维和白色纤维。\n\n步骤 1：确定所需的 蓝色纤维数量。\n这件长袍需要 2 卷蓝色纤维。\n\n步骤 2：计算所需的白色纤维数量。\n这件长袍所需的白色纤维是蓝色纤维的一半。 既然需要 2 卷蓝色纤维，那么就需要 2 / 2 = 1 捆白色纤维。\n\n步骤 3：将蓝色纤维和白色纤维的数量相加，得出所需卷数。\n总卷数 卷数 =蓝色纤维 +白色纤维\n总卷数 = 2 + 1\n总卷数 = 3。
缩短	计算长袍所需的总卷数（蓝色 + 白色纤维）：\n\n步骤 1：需要 2 卷蓝色纤维。\n\n步骤 2：白色纤维是蓝色纤维的一半，因此需要 2/2=1 捆。\n\n步骤 3：总计 = 2+1=3。
扩写	要解决确定制作长袍所需卷数的问题，我们必须同时考虑制作长袍所需的蓝色纤维和白色纤维。 所需的纤维。\n\n步骤 1：首先，我们需要确定制作长袍所需的蓝色纤维的具体数量。查看相关规格或要求后 可知，制作长袍需要 2 卷蓝色纤维。\n\n步骤 2：接下来，我们需要计算所需白色纤维的数量。根据 已知信息，长袍所需的白色纤维是蓝色纤维的一半。由于我们已经确定需要 2 卷蓝色纤维，因此可以计算白色纤维的数量 ，即取这个数量的一半。具体计算如下：2 卷蓝色纤维除以 2，等于 1 卷。因此，制作长袍需要 1 卷白色纤维 纤维。\n\n步骤 3：最后，要计算这件长袍总共需要多少卷纤维，我们需要把已确定的蓝色纤维和白色纤维的用量加起来。 也就是说，将 2 卷蓝色纤维与 1 卷白色纤维相加。计算结果为： 2 + 1 = 3。
移除换行符	要解决这个问题，我们需要确定这件长袍总共需要多少卷纤维，其中包括蓝色和白色纤维。步骤 1：确定所需蓝色 纤维的用量。这件长袍需要 2 卷蓝色纤维。步骤 2：计算所需白色纤维的用量。这件长袍所需的白色纤维是蓝色纤维的一半。由于需要 2 卷蓝色纤维，因此需要 2 / 2 = 1 捆白色纤维。步骤 3：将蓝色和白色纤维的用量相加，得出所需纤维的总卷数。总卷数 = 蓝色纤维 + 白色 纤维。总计卷数 = 2 + 1。总计卷数 = 3。答案：3。
	破坏逻辑的修改
改变数字	要解决这个问题，我们需要确定长袍所需的捆数总量，其中包括蓝色纤维和白色纤维。\n\n第一步：确定所需的 蓝色纤维数量。\n长袍需要4捆蓝色纤维。\n\n第二步：计算所需的白色纤维数量。\n长袍所需的白色纤维只有蓝色纤维的一半。 既然需要4捆蓝色纤维，那么白色纤维就需要4 / 2 = 2 捆。\n\n第三步：将蓝色纤维和白色纤维的数量相加，得出所需的捆数总量。\n总计 卷 =蓝色纤维 +白色纤维\n卷数合计 = 4 + 2\n卷数合计 = 6\n\n答案：6。
更改逻辑	要解决这个问题，我们需要确定这件长袍所需的卷数总计，其中包括蓝色纤维和白色纤维。\n\n步骤 1：确定所需蓝色纤维的数量 。\n这件长袍需要 2 卷白色纤维。\n\n步骤 2：计算所需白色纤维的数量。\n这件长袍所需的白色纤维是蓝色纤维的两倍 。由于需要 2 卷蓝色纤维，因此需要 2 * 2 = 4 卷白色纤维。\n\n步骤 3：将蓝色纤维和白色纤维的数量相减，以求出卷数总计 。\n卷数总计 = 蓝色纤维 - 白色纤维\n卷数总计 = 2- 1\n卷数总计 = 1\n\n答案：1。
添加额外步骤	要解决这个问题，我们需要确定这件长袍所需的卷数，包括蓝色纤维和白色纤维。\n\n步骤 1：确定所需的 蓝色纤维的用量。\n这件长袍需要 2 卷蓝色纤维。\n\n步骤 2：计算所需的白色纤维用量。\n这件长袍所需的白色纤维是蓝色纤维的一半。 既然需要 2 卷蓝色纤维，那么就需要 2 / 2 = 1 卷白色纤维。\n\n步骤 3：将蓝色纤维和白色纤维的用量相加，得出所需的总卷数。\n总 卷数 = 蓝色纤维 + 白色纤维\n总卷数 = 2 + 1\n总卷数 = 3\n\n步骤 4：由于计算错误，从总数中减去 1。\n总卷数 = 3- 1\n总卷数 = 2\n\n答案：2。
篡改最后几步	要解决这个问题，我们需要确定这件长袍所需的卷数，包括蓝色纤维和白色纤维。\n\n步骤 1：确定所需的 蓝色纤维的数量。\n这件长袍需要 2 卷蓝色纤维。\n\n步骤 2：计算所需的白色纤维数量。\n这件长袍所需的白色纤维是蓝色纤维的一半。 既然需要 2 卷蓝色纤维，那么就需要 2 / 2 = 1 捆白色纤维。\n\n步骤 3：将蓝色和白色纤维的数量相加，求出所需卷数总和。\n总 卷数 =蓝色纤维 ×白色纤维\n总卷数 = 2 × 1\n总卷数 = 2\n\n答案：2。
重复问题	为了解决这个问题，我们需要确定制作这件长袍总共需要多少卷纤维，包括蓝色纤维和白色纤维。一件长袍需要2捆蓝色纤维和 一半数量的白色纤维。总共需要多少捆？一件长袍需要2捆蓝色纤维和一半数量的白色纤维。总共需要多少捆？一 件长袍需要2捆蓝色纤维和一半数量的白色纤维。总共需要多少捆？一件长袍需要2捆蓝色纤维和一半数量的白色纤维。总 共需要多少捆？一件长袍需要2捆蓝色纤维和一半数量的白色纤维。总共需要多少捆？

F. AIME题目上的轨迹级置信度动态

图 5 绘制了 *Qwen3-4B-Instruct*在所有 AIME24/25 问题上的轨迹级置信度轨迹。每条曲线对应一条采样轨迹，其中正确轨迹以红色表示，错误轨迹以绿色表示。



(a) AIME24



(b) AIME25

图*5.* (a) AIME24和(b) AIME25中各问题的轨迹级置信度轨迹。

G. 推理正确性估计的细粒度可视化

为了补充主要结果，我们展示了 OTV 正确性估计的词元级可视化，其中生成的推理文本以颜色编码，从绿色（低置信度）到红色（高置信度）。我们纳入了六个示例，涵盖三种比较：1）同一问题的错误轨迹与正确轨迹（示例 1–2），展示了明显的置信度分离；2）较短与较长的正确解答（示例 3–4），较短的轨迹始终保持较高置信度，而较长的轨迹波动更大，表明存在中间不确定性；3）较大模型针对同一问题生成的正确轨迹与错误轨迹（示例 5–6），两种情况下置信度都在接近结尾时下降，但正确轨迹的整体置信度仍显著更高。

示例 1（AIME25，问题23）：Qwen3-4B-Instruct-OTV

区间 $0 < x < 2\pi$ 中有 n 个 x 值，其中 $f(x) = \sin(7\pi \cdot \sin(5x)) = 0$。对于这些 x 的 t 个 n 值，$y = f(x)$ 的图像与 x 轴相切。求 $n + t$。（正确答案：149 ）

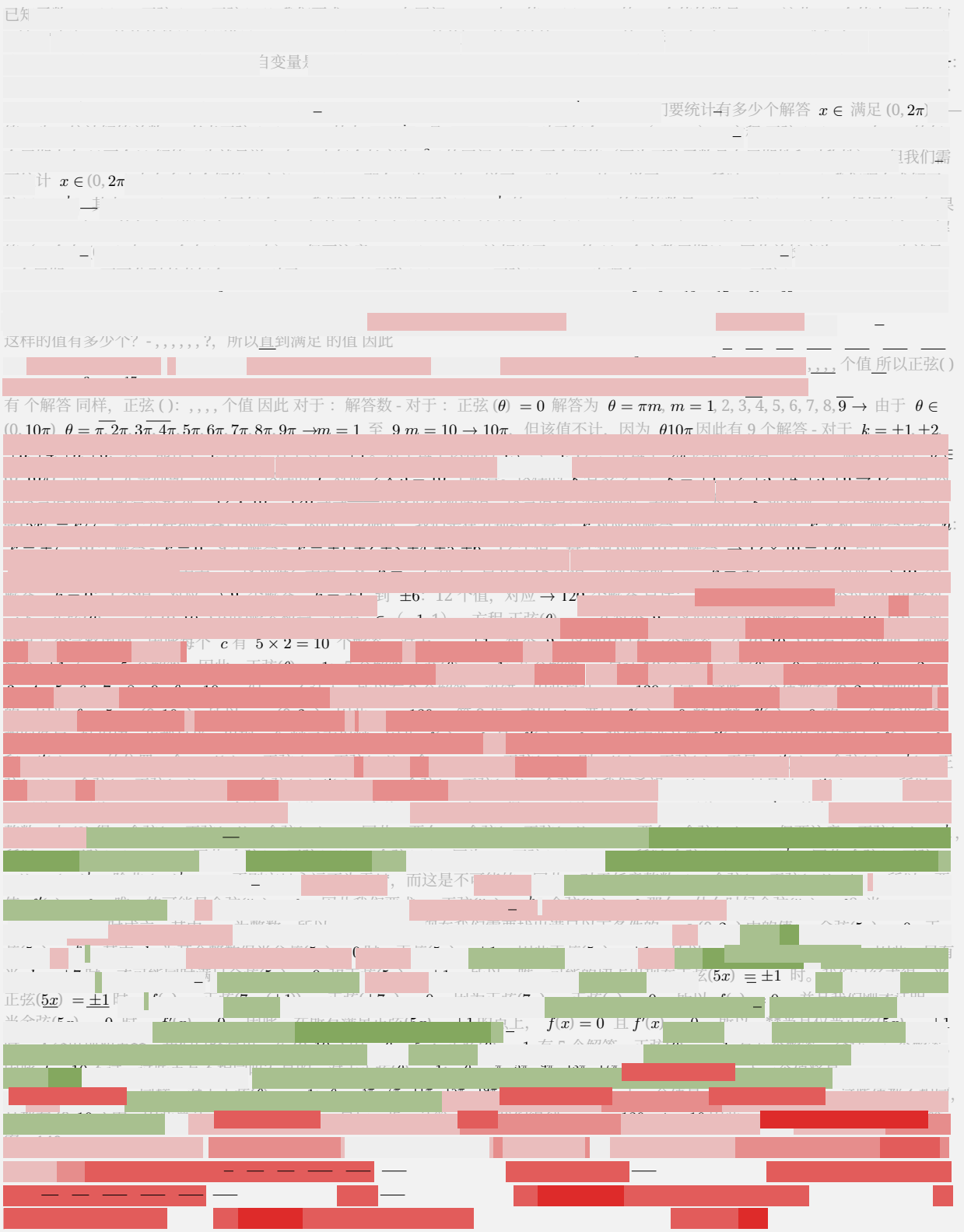


示例 2（AIME25，问题23）：Qwen3-4B-Instruct-OTV

在区间 $0 < x$ 的这些 2π 内，值中的个 t，x 值满足 $f(x)$ 的图像与 $\sin(5x)$ 轴相切。求 $\sin(5x)$（正确答案：149）

已知 [illegible] 自变量是 [illegible] 要统计有多少个解答 $x \in$ 满足 $(0, 2\pi)$ [illegible] 我们需 [illegible] 计 $x \in (0, 2\pi$ [illegible]

这样的值有多少个？ - , , , , , , ?，所以直到满足 的值 因此 [illegible] , , , , 个值 所以正弦( ) [illegible] 有 个解答 同样，正弦 ( )：, , , , 个值 因此 对于： 解答数 - 对于： 正弦 $(\theta) = 0$ 解答为 $\theta = \pi m$, $m = 1, 2, 3, 4, 5, 6, 7, 8, 9 \to$ 由于 $\theta \in (0, 10\pi)$ $\theta = \pi, 2\pi, 3\pi, 4\pi, 5\pi, 6\pi, 7\pi, 8\pi, 9\pi \to m = 1$ 至 9 $m = 10 \to 10\pi$，但该值不计，因为 $\theta 10\pi$ 因此有 9 个解答 - 对于 $k = +1, +2$ [illegible] 到 ± 6：12 个值，对应 $\to 12$ [illegible] 每个 c 有 $5 \times 2 = 10$ 个 [illegible] 而这是不 [illegible] 弦$(5x) = \pm 1$ 时。 [illegible] 正弦$(5x) = \pm 1$ 时 [illegible] 当余弦$(5x) = 0$ 时 [illegible] 在所有满足正弦$(5x) = \pm 1$ 的点上，$f(x) = 0$ 且 $f'(x) = 0$ [illegible] 有 5 个解答 [illegible]



示例 3（AIME24，问题 17）：Qwen3-4B-Instruct-OTV

存在实数 ers x 和 y，二者均大于 1，且满足 $\log_x(y^x) = \log_y\left(x^{4y}\right) = 10$。求 xy。（正确答案：25）



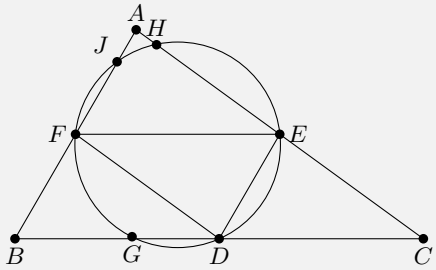
示例 4（AIME24，第22题）：Qwen3-4B-Instruct-OTV

设 $\triangle ABC$ 的外心为 O、内心为 I，且 $IA \perp OI$，外接圆半径为13，内切圆半径为6。求 $AB \cdot AC$。（正确答案：468）



示例 5（AIME25，第19题）：DAPO-Qwen-32B-OTV

假设 $\triangle ABC$的角为 $\angle BAC = 84^\circ$、 $\angle ABC = 60^\circ$和 $\angle ACB = 36^{circ}$。设 D、 E和 F分别是边BC、 AC和 AB的中点。 $\triangle DEF$的外接圆分别与 BD、 AE和 AF相交于点 G、 H和 J。点 G、 D、 E、 H、 J和 F将 $\triangle DEF$的外接圆分成六条劣弧，如图所示。求 $DE + 2 \cdot HJ + 3 \cdot FG$，其中各弧以度为单位度量。（正确答案：336）



要[illegible]
$\angle BAC = 84^\circ$、[illegible]别[illegible]的中点。[illegible]
是[illegible]角形。[illegible]上是[illegible]边的中点[illegible]
还[illegible]
[illegible]接圆是[illegible]我们需要[illegible]度数[illegible]此
我[illegible]
的[illegible]具体来[illegible]的外[illegible]
九[illegible]
Thus, 弧[illegible]的两倍[illegible]，弧[illegible]的两倍[illegible]，且其[illegible]
弧[illegible]是[illegible]的两倍。[illegible]
[illegible]的[illegible]是[illegible]是一条中[illegible]是[illegible]
[illegible]点。要找[illegible]门需要[illegible]解这些[illegible]
F[illegible]线和[illegible]
九[illegible]过[illegible]并与[illegible]
再次[illegible]与圆[illegible]点 J。[illegible]
HJ，[illegible]是由[illegible]因此的度[illegible]现在我[illegible]们用一[illegible]角[illegible]
追踪[illegible]中点，而[illegible]，直[illegible]BD[illegible]
AF[illegible]点 G、[illegible]九[illegible]
中线[illegible]下事[illegible]弧的[illegible]
这些[illegible]和[illegible]由[illegible]位于[illegible]和九[illegible]点[illegible]，因[illegible]
对[illegible]和[illegible]直[illegible]关键的[illegible]察是[illegible]
点[illegible]
以[illegible]有关[illegible]
$\triangle ABC$。[illegible]
以[illegible]
对[illegible]
三[illegible]
交[illegible]的[illegible]点圆。[illegible] $\cdot HJ + 3 \cdot FG$，[illegible]
的[illegible]现在我们需[illegible]在[illegible]
该[illegible]
与[illegible]和[illegible]相关，而[illegible]里。[illegible]
$FG = EF = 168^\circ/2 =$ 与该配置相关的一段弧。将该配置化[illegible]
弧[illegible] $\cdot HJ + 3 \cdot FG$[illegible]
该[illegible]
该[illegible]之间的[illegible]
计[illegible]
为[illegible]些值代入[illegible]
代入[illegible] $\cdot HJ + 3 \cdot FG$.[illegible]
简[illegible]
$FG = 48^\circ$.[illegible] $\cdot HJ + 3 \cdot FG = 72 + 2 \cdot 60 + 3 \cdot 48$。[illegible] $\cdot 60 = 72 + 120 = 192$.[illegible]
项[illegible] $\cdot FG = 3 \cdot$[illegible] $\cdot HJ + 3 \cdot$[illegible] $FG = 336$.[illegible]
$48 = 144$.现在，[illegible]因此，该表达式的值为：$DE + 2$[illegible]

示例 6（AIME25，第19题）：DAPO-Qwen-32B-OTV

设 $\triangle ABC$的角为 $\angle BAC = 84°$、$\angle ABC = 60°$和 $\angle ACB = 36^{circ}$。设 D、E和 F分别为边 BC、AC和 AB的中点。$\triangle DEF$的外接圆分别与 BD、AE和 AF交于点 G、H和 J。如图所示，点 G、D、E、H、J和 F将 $\triangle DEF$的外接圆分成六段劣弧。求 $DE + 2 \cdot HJ + 3 \cdot FG$，弧的度数以度为单位。（正确答案：336）

